KEYFRAME
MAGAZINE

THE PUBLICATION FOR LIGHTWAVE AND CINEMA 4D ENTHUSIASTS

Workplace Wackos

Messiah Muscle II

Motion Designer

Caustics

Deadlines

The Mummy Returns

A Sip of C.O.F.F.E.E.

Advanced Character Setup

Rectifying Reference Photos

Selecting a VFX Training Program

Grzegorz Jonkajtys'
MANTIS

Place with computer magazines

07

0 74470 92596 5

Issue #23 July/August 2001
$9.95 US $14.95 Canada

Why do animators prefer dpsReality?

"I love the easy integration, allowing me to render directly to a DPS file without the need to waste disk space by first rendering to frames, particularly important for long sequences in an animation production."

Fred Heiman
Globe Video Productions, Los Gatos, CA

"dpsReality has increased our company's productivity level dramatically, and time is money! We know we can place our trust in the reliability, reputation, and support DPS provides for our company."

Don Wilcox, CEO and Director
Neutron Films, Sacramento, CA

"Lockstep for 3ds max™ is an integral part of our animation process. Being able to control the playback functions of the dpsReality board from within 3ds max™ saves time and keeps the process flowing."

Mike Truly
Truly Media, Palmer Lake, CO

The dpsReality™ Studio Digital Disk Recorder (SDDR) is the ultimate tool for animators. Featuring compressed and uncompressed video playback and recording with integrated alpha channel support (4:2:2:4), dpsReality provides the best possible video quality for rotoscoping, compositing, and outputting your animations.

The uniquely flexible DPS Virtual Tape File System™, first introduced with our Emmy® Award-winning Perception Video Recorder®, provides seamless and transparent integration between dpsReality, your animation software and all popular image file formats, with full network rendering support. Your software can even render to dpsReality while the hardware is playing back the video file, making it ideal for render farms and collaborative environments. The included LockStep™plug-ins integrate your 3d studio max® and LightWave® 3D animation software with dpsReality even further, allowing you to composite, playback, and output video clips without ever having to leave the animation program's interface.

With a comprehensive range of video and audio I/O and dpsNetStream™ technology for real-time web streaming and MPEG-2 output, dpsReality includes all of the features you need to create and edit your animations and output them for video, broadcast, CD-ROM, DVD and Internet streaming.

dpsReality™

MAKE THE UNREAL REAL

www.dps.com
800-775-3314

DPS®
A LEITCH® BRAND

©2001 Leitch, DPS, the DPS logo and Perception Video Recorder are registered trademarks and think video @ Leitch, dpsReality, dpsNetStream, DPS Virtual Tape File System and DPS LockStep are trademarks of Leitch Technology Corporation. 3d studio max is a registered trademark of autodesk/discreet and LightWave is a registered trademark of NewTek. All other trademarks are property of their respective holders.

the ultimate 3d vector tool

electric rain®

LIGHTWAVE® PLUG-IN

TRANSFORM YOUR LIGHTWAVE® SCENES INTO COMPACT MACROMEDIA® FLASH™ FILES

Swift 3D LW is a vector rendering plug-in enabling 3D designers to quickly generate vector-based animations directly from the LightWave 3D interface.

Render to the Most Vector File Formats

- Macromedia Flash (SWF)
- Adobe® Illustrator® (AI)
- Encapsulated PostScript (EPS)
- Scalable Vector Graphics (SVG)

You take care of the 3D, we'll handle the Vectors!

Choose From the Widest Spectrum of Output Options

- 3 Styles of Outline with Color and Weight Control
- 5 Styles of Cartoon/Cel Shading
- 2 Styles of Gradients including Per-Polygon
- Shadows - Self, Plane and Object Casting
- Specular Highlights - Enhances Cartoons and Gradients

Get a multi-use tool without the annoying infomercial.

Utilize the Fastest and Most Accurate Rendering Engine

- Industry-leading RAViX II Vector Technology inside
- Up to 10 times faster than other 3D-to-vector technologies
- Handles models with more than 200,000 polygons

This is the Monster Truck of Renderers!

Create Completely Optimized Files Without Flash

- Does not rely on outside applications to optimize SWF files
- Fine tune the level of detail and the smoothness of curves

Files so small you'll wonder what went wrong.

Comes With Its Own Preview System

- Completely eliminates unnecessary rendering
- See the goods without leaving the LightWave interface

Click Render - See File - Drop Jaw - Publish to Web.

For more information, visit us on the Web @ www.erain.com

award winning DevLab
http://devlab.swift3d.com.
nce six unbelievable
and witness the potential
3D LW.

KEYFRAME MAGAZINE

Editor
Doug Kelly *dougk@keyframemag.com*

Advertising Rep
Charles Edgin *charles@keyframemag.com*

Co-Conspirators, Writers, and Artists
Cailean Babcock *cailean@provalue.net*
Jared Bendis *jaredjared@earthlink.net*
James Jacobs *jamesj@compt.com*
Mike Jakab *imaginaryforces.com*
Grzegorz Jonkajtys *mantis@grafik.3d.pl*
Patty Kelly *dakelly@megalink.net*
Aaron Matthew *aaron@maxon.net*
Christopher Mills *cmills@lizardlounge.com*
Steve Pugh *steve@foundation-i.com*
Michael Reiser *Mikereiser@aol.com*
Jeremy Squier *j_squier@maxon.net*
Kyle Thatch *kyle@kylethatchstudios.com*
thorn *thorn@maxon.net*
David Warner *dave@tv3d.com*
Ron Wolfe *rwolfejr@woh.rr.com*

The Digital Underground
DMG Publishing
2756 N. Green Valley Pkwy. #261
Henderson, NV 89014
702-990-8656

Subscriptions & Back Issues
Order online at www.keyframemag.com
Email sales@keyframemag.com
1-888-778-WAVE

Keyframe Magazine is published 6 times annually by DMG Publishing. © 2001 DMG Publishing. All rights reserved. Reproduction, in part or in whole, without written permission is strictly prohibited. Printed in the USA.

Printer
Ries Graphics ltd.- A high-tech, award winning printing company in Butler, WI
Ph. 262-781-7520 * Fax 262-781-1842
www.riesgraphics.com
www.riesnet.com

The Secret Formula
Keyframe Magazine is created using Quark 4.1, Photoshop 5.0, LightWave 6.5, and various plug-ins.

No dongles were harmed in the making of this publication.

Departments

3 **Editorial** by Doug Kelly
- Changes

4 **Cover Story: MANTIS** by Grzegorz Jonkajtys
- Animation with a morbid moral

6 **Ear to the Wall**
- News from the VFX arena

8 **On Topic** by David Warner
- Alert Level, Rail Clone, Curve To Motion, Multithreading

10 **Programmer's Hideaway** by Jeremy Squier and Aaron Matthew
- A Sip of C.O.F.F.E.E.

12 **Mac: Up Close and Personal** by Cailean Babcock
- Vector Programs & Animation

38 **Reality Check** by Patty Kelly
- Selecting A VFX Training Program

59 **Reviews** by Kyle Thatch
- Swift 3D

64 **Final Cut** by Steve Pugh
- It Takes All Kinds

Tutorials

15 **Advanced Character Setup Part II** by James Jacobs
- Character Rigging for Arms and Torso

22 **Messiah Muscle, Part II** by Ron Wolfe
- Using Multiple Skeletons for Muscle Motion

29 **Case Study: *The Mummy Returns*** by Mike Jakab
- The Making of the Title Sequence

34 **Caustics** by Christopher Mills
- Diving in with Cinema 4D and NICKL

44 **Motion Designer Nuts and Bolts** by Michael Reiser
- Secondary Motion for Character Animation

47 **Architectural Photography** by Jared Bendis
- Creating Perfectly Flat Maps

54 **Beating the Broadcast Deadline** by thorn
- Broadcast Titles

Issue #23 July/August 2001

CHANGES

By the time you read this, most of the visual effects community will already have heard the news that Keyframe has expanded our focus. We are now covering products marketed by MAXON, including Cinema 4D and BodyPaint 3D, as well as continuing to cover NewTek's LightWave, Aura and Video Toaster. As always, we also publish tutorials, reviews and news on any compatible product, such as project:messiah or Swift3D.

To date, most of the response to our expanded focus has been positive. A few readers have worried that LightWave coverage will suffer. I'd like to do my best to reassure all our readers that we are truly *adding* coverage, not dividing it. I believe the magazine you're holding adequately addresses this concern: Keyframe is now 64 pages, twice our last issue's count, and the plan is to keep that number. We're still working on building up Cinema 4D coverage, so the balance of tutorials in this issue is slightly in LightWave's favor. However, there are a number of feature articles and tutorials that apply equally well to *any* 3D software.

An appropriate keynote to this first expanded issue is that the cover image was created by artist Grzegorz Jonkajtys for his short film *MANTIS* using both Cinema 4D *and* LightWave. I don't know any working artist who uses one software tool exclusively; just as a painter would be ill-advised (not to say insane) to use only one type of brush, digital artists would do well to keep a reasonable selection of compatible tools in their kits. I believe Keyframe is serving our readers best when we independently and dispassionately examine whatever tools are necessary to create the art.

I'm also pleased to report a continued increase in the sophistication of our readers and our contributors. Reader feedback indicates that we are consistently achieving our goal of providing tutorials and case studies that are of use to both working professionals and not-quite-pro-yet students and hobbyists.

Based on the strongly positive feedback on last issue's 'Reality Check' feature article (Patty Kelly's "Living With An Animator"), we are trying it out as a regular column. This will be a resource for readers dealing with the world outside the studio, covering life and business issues that are important to us all. In this issue, Patty takes a look at criteria and questions for those of you considering formal education in visual effects.

As always, we appreciate your feedback. Emails and letters are always welcome, and if you're browsing this issue at SIGGRAPH, please stop by Booth 2413 and say hello!

Thanks for reading,

Doug Kelly

MANTIS
a short film by grzegorz jonkajtys

concept, design and story,
animation, modeling, lighting and texturing:
grzegorz jonkajtys

music:
piotr lichwierowicz

software:
Cinema4D XL 6.1
Lightwave 6.5
Adobe After Effects
Adobe Photoshop
Adobe Premiere

nkajtys all rights reserved

www.mantis.3d.pl

You can download the movie from links on my website:
http://www.www.mantis.3d.pl/

The Fast,
Powerful Production
Tool for Animation,
Video Paint

MENTION THIS AD AND
GET 5% OFF NEWTEK SOFTWARE!

CHECK OUT OUR GREAT PRICES!
1-800-944-5520 OR 1-702-990-2277
www.dynamic-realities.com

WE'RE TAKING ORDERS FOR
VIDEO TOASTER 2

OF COURSE
OUR PLUGINS
ARE ALWAYS READY TO SHIP!

NATURE FX
REALITOOLS
LUMETOOLS
NAPALM NEW!
IMPACT 2 NEW!

Dynamic Realities

News from the VFX arena

Right Hemisphere® Announces Enhanced LightWave 3D® Support

LightWave 3D users can now experience the full power of Deep Paint 3D and Texture Weapons directly

Right Hemisphere® announced that it will ship the latest Deep Paint 3D® V1.61 with direct plug-in support for NewTek's® LightWave 3D® Version 6.5. A new plug-in for LightWave 3D has been developed by Right Hemisphere that allows a direct connection to be established between Deep Paint 3D and LightWave 3D software. This creates a very smooth texturing workflow for LightWave 3D artists. Users can model and create UV mapping in LightWave 3D and then link directly to a complete 3D texturing and painting solution without loading and saving separate 3D projects.

The direct link from LightWave 3D to Deep Paint 3D allows artists to model in LightWave 3D and then directly link to Deep Paint 3D to paint directly on the 3D model seamlessly, crossing complex seams and multiple materials in one brush stroke. They can then immediately switch to LightWave 3D for further modeling or animation work.

All types of UV mapping are supported including Spherical, Cylindrical and Planar as well as the more advanced types such as Atlas mapping. Multiple rendering channels such as Color, Bump, Glow, Shininess and Opacity are supported and can be brushed on simultaneously and passed seamlessly back and forth between the two applications.

In addition, Deep Paint 3D can link LightWave users to Adobe's ® highly respected Photoshop® painting package. Users can then use Photoshop to paint unwrapped or projected textures while linked to Deep Paint 3D and LightWave 3D.

"The Deep Paint 3D V1.61 LightWave Plug-in has been an incredible asset to our art pipeline. LightWave's Texture Atlas and Deep Paint 3D's Projection Paint are all an artist needs. I'll never go back to taking screen shots of wire frames, dealing with stretching, and toying with UVs. It's nice to just select geometry and press "Paint". Easy." says Vance Kovacs, Lead Artist of Black Isle Studios.

"We at NewTek are very excited about the direct link from

Continued on next page

KEYFRAME MAGAZINE EXPANDS FOCUS TO INCLUDE MAXON PRODUCTS

NEWBURY PARK, CA - July 9, 2001 - The publishers of Keyframe Magazine today announced the periodical's focus will be expanded to include MAXON's award-winning professional 3D solutions, CINEMA 4D XL and BodyPaint 3D. Geared towards the professional 3D artist, Keyframe has been renowned for its high-end tutorials and coverage of production techniques for LightWave users. Commencing with the next issue of the magazine (July/August 2001), users of MAXON products will benefit from the same high-quality techniques and articles.

MAXON's products have found acceptance among 3D artists throughout the content-creation industries, increasing the opportunity to profile professional users and share high-end production techniques.

"Keyframe Magazine has averaged 20,000 words per issue of tutorials, product reviews and news; the most in-depth, professional tutorials of any 3D magazine," states Keyframe editor Doug Kelly. "We will maintain that coverage for LightWave, and will add comparable coverage for MAXON's CINEMA 4D XL and BodyPaint 3D by expanding our page count. We will continue to cover third-party products that work with NewTek — and now MAXON — products, including plug-ins, hardware, utilities, and complementary tools. MAXON's own BP<->LW Exchange plug-in enables BodyPaint 3D to work with LightWave objects, and CINEMA 4D XL works with LightWave files. This cross-compatibility demonstrates that Keyframe Magazine's new coverage is a logical and useful expansion for all our readers."

"We have always respected Keyframe Magazine and the guidance it provides. The value to our customers will be unparalleled," said Paul Babb, president and CEO of MAXON Computer, Inc. "We're thrilled that the publishers have chosen to include MAXON's products in their coverage."

About Keyframe Magazine
Keyframe Magazine is a bimonthly publication of DMG Publishing. Since January 1997, Keyframe (formerly known as LightWavin') has published in-depth tutorials, product reviews and news for the computer-generated effects community.

About CINEMA 4D
CINEMA 4D is a fully integrated 3D-modeling, animation and rendering package for Macintosh and Windows. It is well known for its surprisingly fast and photorealistic-quality raytracer. For more information or to download a free demo version, please visit MAXON's web site at http://www.maxoncomputer.com.

About BodyPaint 3D
BodyPaint 3D is an advanced 3D painting application for creating high-quality detailed textures. BodyPaint 3D seamlessly integrates into CINEMA 4D XL.

Ear To The Wall (continued)

Deep Paint 3D to LightWave 3D," said Brad Peebler, NewTek's vice president of 3D graphics. "Many of our users are already using Right Hemisphere's products, and this direct plug-in will make Deep Paint 3D an obvious choice for artists using LightWave. The time and effort that this combination will save will allow for better and faster graphics creation and a more efficient workflow."

"We are very pleased to be working with Newtek to create a complete 3D texturing solution for all Lightwave 3D users. The combination of Deep Paint 3D and Lightwave 3D will bring great joy and productivity to 3D artists!" says Michael Lynch, Right Hemisphere CEO.

Right Hemisphere's Deep Paint 3D® has already become the standard for 3D painting & texturing for 3D content creators. This latest feature addition makes texturing within LightWave the most fun thing to do in the creative process.

Deep Paint 3D 1.61 is due to ship in July 2001. More information on Deep Paint 3D and Texture Weapons is available at www.righthemisphere.com .

About Right Hemisphere

Right Hemisphere www.righthemisphere.com is committed to enriching the human experience through software products and services, creating unmatched value and performance for digital artists. The company partners with many key industry leaders such as Intel, nVidia, discreet, Softimage, Newtek, Alias|Wavefront and Adobe. Winners of numerous industry product awards, the company has a centre of technical excellence for graphics programming based in Auckland, New Zealand. Right Hemisphere is continually innovating to develop systems for the production, management, protection and commerce of 3D content; in the rapidly growing markets where technology, communication and art converge.

The privately funded company is based in Auckland, New Zealand, and has incorporated a US subsidiary, Right Hemisphere USA, to manage North American sales and distribution. Right Hemisphere can be contacted on US Toll Free PH +1 877 309 3204 and on Intl' PH +64 9 523 46 70.

About NewTek

With headquarters in San Antonio, Texas, NewTek is a leading provider of full-featured video editing, animation and special effects tools Video Toaster, LightWave 3D and Aura. NewTek launched the desktop video revolution in 1990 with the release of the legendary Video Toaster. The company's products are used worldwide on projects ranging from home video to feature film. Recent film and television projects using NewTek software include Charlie's Angels, How The Grinch Stole Christmas, 6th Day, Red Planet, X-Men The Movie, Pitch Black, Dracula 2000, Roughnecks: Starship Troopers, Star Trek: Voyager, Star Trek: Deep Space Nine, The X-Files, Ally McBeal, Dune: The Miniseries, Dawson's Creek and Family Law. Contact NewTek at 5131 Beckwith Blvd, San Antonio, TX, 78249. 210-370-8000, www.newtek.com or www.lightwave6.com.

Pre-Publication Offer

Coming September 2001

LightWave™ Applied

version 6.5

NEW from the industry-leading experts that brought you *LightWave 3D™ Applied, Version 5.6...*

Imagine a tutorial that has 100 steps and more than 50 images, and that's just one chapter! *LightWave™ Applied—Version 6.5* teaches readers from scratch how to master LightWave™ 6.5 through the development of actual projects. Readers will be treated to in-depth tutorial chapters on using LightWave™ 6.5 features to build items like buildings, cars, mountains and more. Readers will learn top cel shading techniques to make a 3D character look hand drawn when rendered. Also covered is how to prep a character for animation. From IK and Bones techniques to creating an entire architecture landscape with camera fly-throughs, *LightWave™ Applied—Version 6.5* gives in-depth coverage to techniques critical for animation success.

Excellent guide for instructors teaching computer animation.

Features include:

- Ten step-by-step tutorial chapters to help you construct detailed and realistic objects, many you'll learn to animate
- A specially designed Website, with two bonus chapters, to help answer any questions
- A FREE CD-ROM packed full of material from the book along with bonus items such as an updated version of "Breaking into Hollywood."
- A special introduction section for beginners to LightWave™
- A special chapter teaching readers how to start their own local animation company
- Insight from industry leading authors along with their tips and secrets to help you succeed

$\$44^{95}$ if purchased before September 15th

$\$49^{95}$ Regular Price,
Over 750 pages, Soft Cover, Item #POSB021,
FREE CD-ROM

Call 1-800-598-6008
Outside the U.S. 1-218-723-9180
Fax 218-723-9146
Visit and order on-line at
www.advanstarbooks.com/lightwave/
Please mention code O-KF when ordering

Alert Level, Rail Clone, Curve To Motion, Multithreading

by David Warner

Another issue, another trade show. This time we have SIGGRAPH 2001 and for those of you who are lucky enough to be attending, you already know how much fun SIGGRAPH can be!

SIGGRAPH is traditionally the show for 'artists,' demonstrating tools that focus more on the creative aspect of multimedia production. In other words, SIGGRAPH is the *fun* and *exciting* show for us creative types! The Los Angeles show is extra special because most of the major Hollywood effects studios are close by, and you'll find many of the most talented people in the industry walking the exhibition hall or doing product demos or just showing off their work.

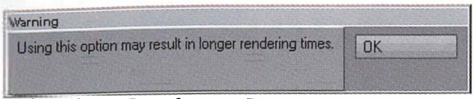
The past two SIGGRAPH shows have unveiled LightWave versions 6.0 and 6.5, so I hope that NewTek will continue this trend and give us a glimpse at the upcoming v7.0. Seeing as how NewTek has provided LightWave 6.0 owners with two free upgrades already (versions 6.5 and 6.5b) I seriously doubt that we'll be seeing a free v6.5c upgrade, but that's quite all right with me. NewTek gave us some major improvements in version 6.5b and I will happily pay a reasonable upgrade fee for the next new version. Of course, if NewTek does offer another free upgrade, well, I won't complain.

So if you are reading these words on the flight home from L.A. then you already know about all the cool new announcements and you certainly don't need me to speculate further about what may or may not happen at SIGGRAPH. If you're sitting in a corner of the exhibition hall somewhere, taking a break from the din of activity on the show floor, then stop wasting time reading my ramblings and go exploring! There's miles of show floor to cover and only a few days to do it!

Here's this issue's selection of questions and answers from the LightWave mailing list...

Q When I go to the Rendering Options panel and turn on the "Show Rendering in Progress" option, I keep getting a warning message that tells me my render times may increase by activating this option. Why is this true and is there anything I can do to get rid of this warning message?

A The only time this warning message (Figure 1) will appear is when you have the Auto-Advance option turned on in the Rendering Options panel and you try to turn on the Show Rendering Progress option. The reason this warning appears is pretty much explained in the warning message itself; by forcing LightWave to display the rendered image as it is actually rendering, you eat up a good number of CPU cycles. This will significantly slow down the render speed of your computer. It is like asking an artist to provide you with an updated preview of his painting every time he makes a brush stroke, thus slowing down the whole process of creating the painting in the first place.


Warning

Using this option may result in longer rendering times.

Figure 1: *The Show Rendering Progress warning message.*

The Show Rendering Progress option is very handy when you want to keep track of how individual frames are rendering as you are actually setting up an animation. However, as soon as you are ready to render the whole animation, this option should definitely be turned off. If you want to be able to keep track of individual frames as they are rendered, you might consider turning on one of the display options that will show you a full frame after it has been rendered. This would allow you to see each finished rendered frame in full size and it will not slow down the rendering process.

When you get to the experience level where you feel that the various warning messages are just an annoyance and you don't want to see them anymore, you can go into the Layout configuration file and edit a line of text that will decrease the number of warnings or eliminate them altogether. If you search your system for a file named "lw3.cfg" and then open that file with your favorite text editor, you will find a number of variable settings that control how LightWave works. Look for a line near the top of the config file that starts with "AlertLevel" and next to it you will most likely see the number zero. By changing this value from zero to one you will decrease the number of warnings and alerts that you receive while using LightWave. By changing this number to two, you will eliminate all warnings and alerts altogether, which is something I don't recommend you do, because many of the warnings are handy reminders to prevent the user from doing something stupid. I keep the AlertLevel value set to one, just to avoid doing too many stupid things. Once you have changed this variable, save the file and the next time you launch LightWave, you will notice fewer warning messages. One important point to remember is that you can only edit the LightWave config files when LightWave is not running. If you make any changes to the config files while LightWave is running, those changes will be erased when you exit from the program because LightWave updates the config files each time it is closed down.

Q Is there an easy way to uniformly distribute a number of cloned objects between two points? For instance, if I wanted to create a strand of pearls with each pearl offset a precise amount from its neighbor, how would I do this?

A This can easily be accomplished by doing a Rail Clone operation along a spline curve, but first you need to know the diameter of each object so that you can come up with an appropriate length for the spline curve. In the example given, if each pearl were 5 centimeters in diameter and you wanted to create a strand of 100 pearls, then you would need to create a spline curve that is 500 centimeters (or 5 meters) in length.

Now that you know how long the spline curve needs to be, you can generate the curve by creating 2 points that are 5 meters apart from each other and then use the Make Open Curve tool to turn these points into a straight spline curve. If you were going to animate the pearls moving around or something, then this would be the ideal way to model the object so that a bone skeleton could be easily added later. However, if you would prefer to model the pearl strand as a closed loop then you would need to create a "closed" spline curve that has more than 2 points.

Once your spline curve is created, you would go to a different layer and create the object that you wish to clone and place the center of that object right over the first point in the spline curve. Activate the Rail Clone tool in the Multiply panel. In the Rail Clone panel (Figure 2) you would set the Segments parameters to Knots, Uniform, and enter a value of 100 for the number of cloned segments. The Knots option tells Modeler to define 100 different points (or knots in the curve) where the object will be cloned and the Uniform option tells Modeler to distribute the clones evenly along the length of the spline curve. You would also want to have the Oriented option turned on if you were making a closed loop of pearls and you wanted the cloned objects to conform to the shape of the spline curve. The Scale option is handy when you are using multiple spline curves to control the shape and size of your cloned objects.

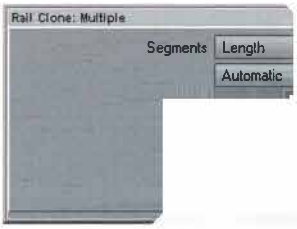


Figure 2: *The Rail Clone options panel.*

Q I'm trying to make my camera follow a path that was created with the Curve To Motion plug-in in Modeler, but the resulting motion path of the camera is very jerky and inconsistent in its movements. Is there any way to do this?

A Unfortunately, this is something that is quite difficult to do in LightWave and has been for many years now. The main reason this is such a difficult thing to do is that the Path-To-Motion Modeler plug-in creates keyframes based upon the location of knots (or points) in your spline curve. Many times you will have some knots bunched together and other knots spread out along the shape of the curve and this has a significant effect on the resulting motion path. As I said, it is very difficult to overcome this big shortcoming in LightWave but here are a few tips to improve your results.

The most important step is to make sure that the knots in your spline curve are distributed evenly along its length before you convert the curve to a motion path. This can be done fairly easily by following a technique that is similar to the answer of the previous question.

First you create the spline curve that you want to turn into a motion path. Copy the first point in that motion path, go to a new layer, and then paste that point into the new layer. Now select the background layer with your spline curve in it and as described in the previous question's answer, select the Rail Clone tool and apply the same options to this clone operation. Be sure to specify an adequate number of knots to ensure that your new spline curve will be nice and smooth. It may even be necessary to specify a large number of knots that is close to the number of frames you will have in your animation. For example, if your animation is going to be 120 frames long, it may be necessary to create 60 knots in your new spline curve.

Now that you have a new set of points created, you must turn them into a new spline curve. Carefully select them one by one, in the order of their path, and then make them into a new curve. Now you can run the Path-to-Motion plug-in and be fairly certain that the resulting motion file will have a uniform velocity to it, making your camera or object follow the motion in a smooth motion.

If you are making the camera follow this new motion path, then you will most likely have problems with the camera just moving in a very unnatural way, which will probably manifest itself as motion sickness for anyone viewing your final rendered animation. One way to get around this is to give your camera a bit of a "look ahead" effect so that the camera anticipates movements into the turns and curves. This can be done very easily by adding a target object, like a simple null object, for the camera to focus on. Apply the same motion path to the null object, set the camera so it is targeted at the null object, and then shift all of the keyframes for the camera's motion path so it is a few frames behind the null object. The appropriate offset value depends on how much you want the camera to look ahead. An offset of 5 frames is a good place to start.

If you are still unhappy with the results of your object or camera's movements then the best alternative may be for you to purchase one of the many fine motion control plug-ins that are available commercially. Worley Labs has a number of good motion control plug-ins that are part of some great plug-in packages. There are many other motion control plug-ins available so if you decide to go this route then get on the net and do some research to figure out which plug-in will meet your needs and fit into your budget.

Q I just installed LightWave on a dual-processor system and I don't see any increase in render speed. What am I doing wrong?

Continued on Page 61

A Sip of C.O.F.F.E.E.

PROGRAMMER'S HIDEAWAY

by Jeremy Squier and Aaron Matthew

C.O.F.F.E.E. is MAXON's internal programming language in which plug-ins and expressions for CINEMA 4D can be written. C.O.F.F.E.E. resembles C/C++ and Java - some of the syntax will be exactly the same and the other parts will be specific to CINEMA 4D. It is very easy for C/C++ and Java programmers to become familiar with C.O.F.F.E.E., which allows you to modify many things in CINEMA 4D that you wouldn't otherwise be able to touch. Of course, if you want to work in a caffeine-free programming environment, you can always program plug-ins through the C++ SDK as well.

What is Code?

Code is a list of instructions a computer executes to do tasks, which can range from adding numbers together to rotating your favorite object. Computers are stupid, meaning they can't learn. A programmer must supply them with code to get them to do anything. The job of a programmer is to write these instructions to make the computer do new and useful things.

Remarks are Not Code

Remarks are words that are not read by the compiler. They will be ignored by the computer at execution time. You can remark all text after the symbol // on a line. For example:

```
/This is a remark
ar a = 4;   //set a equal to four
ar b = c + a / 4; //this might be the right algorithm
```

Remarks are generally used for programmer notes. They can be used for other things as well, but the important thing to realize is that the compiler will ignore them.

Variables, the C.O.F.F.E.E .Way

A feature that might be unnerving for any C programmer is the fact that C.O.F.F.E.E. is type-less, meaning that any variables you declare have no type. All you have to do is declare the fact that you do indeed have a variable by using "var." For instance:

```
var number = 5;
var name = "Bob";
var object;
```

Type-less is a bit of a misnomer; a better description would be type full. Variables in CINEMA 4D can hold any type of data. A variable could hold a number at first, and then you could assign it to contain a string. This versatility also has some pitfalls, but those are beyond the scope of this introduction.

Even though a variable can hold any kind of value (a number, a cube or a matrix), you can still figure out what it is holding at any moment. The **typeof()** function allows you to find this information. The typeof() figures this out by looking at what is contained in the variable. Once it has identified what the type is, it tells you by its return value.

C.O.F.F.E.E. Functions

Functions are named lists of commonly repeated instructions. For instance, you might make a function to "Walk The Dog." Once you write the function, you can just say "Execute the "Walk The Dog" function." This execution is also called 'calling a function'. The advantage is that you need not write the instructions of how to walk the dog every time you want to do so, you just call the function "Walk The Dog" and the computer executes these instructions.

Functions in C.O.F.F.E.E. are prototyped as:

```
function(parm1, parm2, parm3);
function(x, y, z);
function(cube);
function(rock, scale);
walk_the_dog(dog, leash);
```

A prototype is a declaration of what a function is going to look like, so C.O.F.F.E.E. knows how to use the function even before you actually define the function's contents. This allows you to use a function that is defined later on in the code.

Returning from a Function and Return Value

Simply use **return** to return control back to its calling function. Alternatively, you can have a variable or even a number after it:

```
return;     //return to calling function
return 0;   //return to calling function passing back a 0
return a;   //return to calling function passing back a
return a+b;//return to calling function passing back a+b
```

C.O.F.F.E.E. Flow Control

Conditional expressions are used in conditional execution. The following symbol chart lists the logical operators used.

```
== equals to                      1 == 1     is TRUE
!= not equal to                   1 != 1     is FALSE
<= less than or equal to          1 <= 1     is TRUE
>= greater than or equal to       1 >= 1     is TRUE
 < less than                      1 > 1      is FALSE
 > greater than                   1 < 1      is FALSE
```

C.O.F.F.E.E. Conditional Execution

Conditional execution allows certain code to be executed and other code to be skipped based on a certain condition. Following are two examples, the If Statement and the Switch Statement.

If Statement:

```
code_line1;     //always do this
code_line2;      //always do this
if (condition)    //if condition is TRUE
{
        do_this;    //do_this
}
else            //if condition is FALSE
{
        do_that;    //do that
}
code_line3;     //always do this
```

Switch Statement: The switch statement simply checks the value of the variable against a case. If they match it will execute that case.

```
switch (variable)
{
        case 0:
                variable_is_0;
                break;
        case 1:
                variable_is_1;
                break;
        case 2:
                variable_is_2;
                break;
        case 3:
                variable_is_3
                break;
        .
        .
        .
        default:
                variable_is_something_else;
                break;
}
```

Loops are a special type of conditional execution. Following are three examples of loops, the While Loop, the Do-While Loop and the For Loop.

While Loop: while the condition is TRUE do_this.

```
while (condition)
{
        do_this;
}
```

Do-While Loop: do_this and then check if condition is TRUE, if so loop.

```
{
        do_this;
} while (condition);
```

For Loop:
The following example would print out the numbers from 0 through 9.

```
for (initialization; conditional; post-body-
execution)
{
        do_this1;   \
        do_this2;   } loop body
        do_this3;   /
}
```

A For Loop can be re-written for more clarity.

```
initialization;
while (conditional)
{
        do_this1;
        do_this2;
        do_this3;

        post-body-execution;
}

for (i=0; i<10; i++)
{
        println(i);
}
```

Flow Control within Loops

Within a loop you may need more control. There are two keywords that help you here: **continue** and **break**. Continue causes execution of the body to skip to the conditional statement of the loop. Break completely terminates the loop. Also, break applies to switch statements. In this case, it terminates execution of the switch statement and brings execution right to the code after the switch block.

Classes in COFFEE

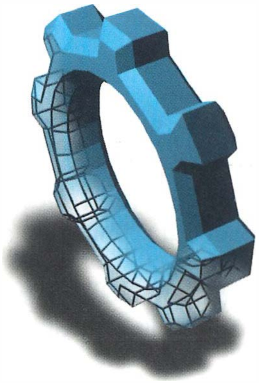
Classes in C.O.F.F.E.E. act pretty much the same way they are explained in any C++ reference or as described in our Coding Primer Tutorial at
http://www.plug-incafe.com/developer_main.asp
The concept of a class is a bit beyond this introductory article. Going through the Coding Primer Tutorial will give you a much better introduction.

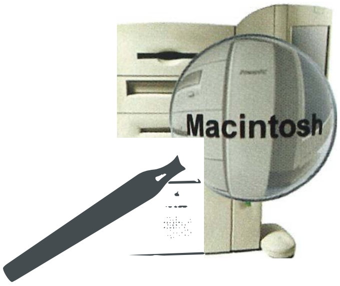
C.O.F.F.E.E. References

A reference for C.O.F.F.E.E. is provided at Plug-in Café
http://www.plug-incafe.com
This is a good source of C.O.F.F.E.E. information. A good place to learn more is reading the previously-referenced Coding Primer Tutorial.

*Aaron Matthew (**aaron@maxon.net**) and Jeremy Squire (**jeremy@maxon.net**) can be contacted through their email addresses or through MAXON:*

MAXON Computer, Inc.
2640 Lavery Ct. Suite A
Newbury Park, CA 91320
805-376-3333 • FAX 805-376-3331





Up Close and Personal

by Cailean Babcock

If you were to name off the ten most important weapons in the arsenal of the self-sufficient animator, you probably wouldn't find "illustration program" on the list. In the realm of broadcast and film, vector-based artwork created in such programs as Adobe Illustrator, Macromedia Freehand, and Corel Draw might seem pretty out of place. That being said - I'm now going to convince you otherwise.

As Mac users, we all know that the Macintosh has been considered the number-one system for 2D graphics creation since the advent of desktop computing. One of the main reasons for this was a company called Adobe, and its first offering on the Macintosh: a little program called Adobe Illustrator.

How on earth can an illustration program help you create sophisticated animations in LightWave? How about logos, advanced texturing effects, model templates, industrial visualization, and forays into the fertile field of multimedia design? Read on!

Logos

First and foremost, an illustration program can be an invaluable tool for managing client logos. If you've ever done a logo animation for a client, you've run into the problem of just how you're going to get a hold of a legible copy of their logo to work with. I'm sure we've all heard the following at least once: "Can't you just scan a copy of it off of my business card?" or even worse, "Why not just download the [incredibly tiny and pixilated] one off of my web page?" I don't know about you, but I usually find the subsequent two hours spent attempting to reproduce a clean, digital copy of that logo on my computer to be an absolute pain (not to mention that it's time better spent actually creating a dynamite animation). On the bright side, if you're billing by the hour, I suppose it's not all bad. If a client already has a logo that is being used in his print publications, dimes to dollars an EPS file exists somewhere containing that logo. Once you've acquired that file, you'll discover several beneficial options now at your disposal:

1. Resolution and File Size. You now have a clean, resolution-independent copy of the logo that takes up virtually no space.

2. Color. If the logo is in color, you can now easily sample the original print colors, sans compression artifacts, and incorporate them into your surfacing in LightWave, if the situation requires.

3. Layer management. The current offering of vector-based illustration programs out there offer similar layer management options as in Photoshop, allowing you to sort out the various elements of your client's logo into discreet items that you can then export into separate files if need be.

4. Serious control over fonts and type. Options such as kerning, tracking, and leading are much more easily controlled in these types of programs. Also, the text can always be re-edited, yet retain the properties of the original type.

5. Better file compatibility. Sometimes it's easier to not have to explain to a client how exactly to go about exporting something into another format to be compatible with you. If you have a program already capable of reading that format, you're already halfway there. One of the nicer perks of Adobe Illustrator, for example, is the ability to access vector artwork already embedded into an Acrobat PDF document (one of the few cases where getting a client logo off of the web can be a good thing).

Texturing

This one is actually a lot more straightforward than it might sound at first. When creating surfaces that involve highly technical detail, such as paneling and rivets on vehicles, a good illustration program can be a real lifesaver. The method is very simple: just use your normal texturing practices of taking a snapshot of the model into an editing program, except this time, use the illustration program. Want to create a line of rivets? It's as easy as drawing a Bezier line and setting a dash value to create a series of rounded dots. Export the results to an image file, and within only a couple of minutes, you're finished. Do the rivets not follow the contour precisely? Go back and edit the curve, and voila, the points are redrawn and ready to generate a new texture. You can easily afford to keep several iterations of the original files, too, due to their small size. As with many of your standard pixel-based image editors, illustration programs offer sophisticated layer management to allow you to manage your image data (see figure 1).

Modeling Templates

You can lay out complex and surprisingly sophisticated modeling templates within a vector program. Because they're completely curve-based, this provides you with very flexible editing options for later modification. Yes, LightWave now provides Bezier curves for advanced object editing. However, this latest offering from NewTek is still in its beginning stages and has a few limitations. The object-creation functions in vector-based editing programs are more robust and sophisticated than in LightWave. Despite LightWave's excellent ability to encompass many modeling paradigms, sometimes you need a tool that is specialized to the task at hand. Once again, layer management is similar to that which you're already used to

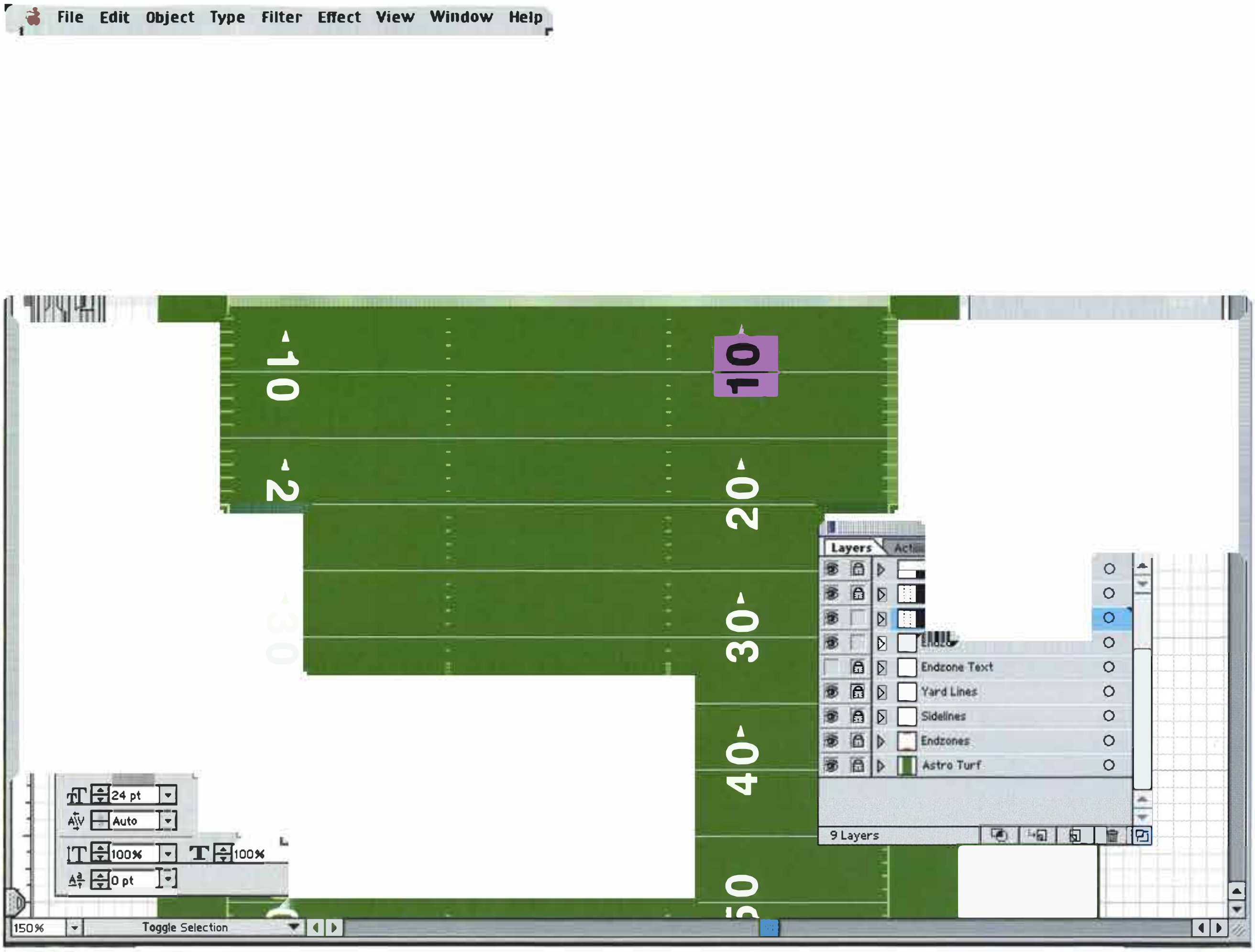


Figure 1: With advanced text control and Photoshop-style layer management, Adobe Illustrator 9 makes creating complex technical textures a snap.

using within Modeler and Photoshop, and with the ability to not only maintain true Bezier curves for creation and later modification, you're job will be that much easier. When it's time to use the EPSF_Loader plug-in to import your object, you'll find that the ability to dial in the resolution of your polygon edges is a real time-saver. You can generate low- and hi-resolution versions from the same file with just a few mouse clicks, saving valuable time later.

Industrial Visualization

This one seemed to deserve a paragraph of its own. If the topic isn't self-explanatory, you could also label this one "CAD/CAM compatibility." One of the newest and fastest-growing sectors of the 3D market is visualization, whereby the animator is hired by a client to present architectural and other industrial design concepts in a photo-realistic fashion that, as the name implied, aids in "visualizing" the final product. Whereas LightWave can import generic CAD formats directly, I have discovered through practice that sometimes these files can contain errors or otherwise be so unwieldy that LightWave can't import the data. It was the ability of the CAD software to export into EPS format that saved the day. I was able to fire up Adobe Illustrator and clean up the file, and (once again) separate the data into layers for easier management within LightWave. In this particular example, being able to delete extraneous information before importing it into LightWave, such as the 600-plus seats in several classrooms of the three-story training facility I was working on, was a real time-saver.

Multimedia Design

If you need a way of ensuring a steady and plentiful flow of projects, there's no better field to enter into now than multimedia design. From web design to DVD distribution, there is a dazzling array of possibilities for content creation. Illustration programs are the first place to start for serious design work for the Flash and Shockwave formats, which have firmly ensconced themselves as the de-facto standard for high-end multimedia sites on the web. Macromedia builds hooks aplenty into Freehand 10, their latest vector-based offering. One of the most innovative features is the ability to construct, preview and export simple Flash animations directly from within the program.

Another option is multimedia CDs for corporate identity brochures, as well as the rapidly expanding market of DVD authoring. With the prices of authoring equipment rapidly dropping, many more content creators can step into this market with a relatively small investment.

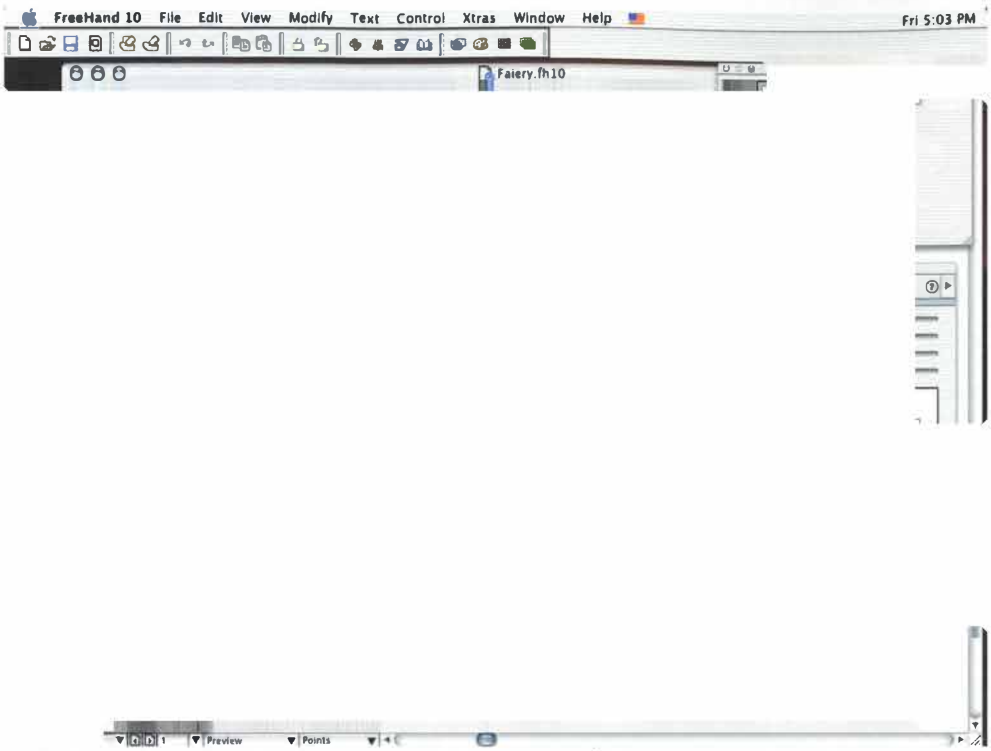


Figure 2: *New and improved for OS X, Macromedia does it again by being the first to bring their highly-acclaimed illustration to Apple's latest operating system.*

Interface design and layout can be a snap using an illustration program. Yes, you can use Photoshop to do the exact same thing. However, vector-based means resolution independent, which can be a real time-saver when you decide you need to change screen sizes for your project. Think of such interface mock-ups as the animatics of multimedia design. There's also no reason that you can't create finished pieces with the program, either; there are an amazing multitude of examples out there of innovative designs that don't incorporate a single "real" pixel.

Software Availability

There are three-and-a-half major applications out there for serious vector-based editing today (I'll explain the half in just a minute). First of all come the tried-and-true staples of Adobe Illustrator, Macromedia Freehand, and Corel Draw. All of these offer EPS file compatibility and mature editing features developed over the last decade. If you're not certain you want to jump in both-feet-first right away, Corel Draw 8 LE for the Macintosh is a free download from Corel's website, at http://www.corel.com/draw8mac_le/download.htm. Each application has its own special set of advantages, and the stiff competition between the companies that create them means frequent updates and many exciting new features for the end-user.

Now on to the "half" in three-and-a-half: with the release of Adobe Photoshop 6, Adobe has incorporated some advanced vector-based editing features that rival its own sister application, Adobe Illustrator. While not as feature-rich as Illustrator, these tools can still offer enough high-end vector-editing features to keep your average animator very happy. Also, Photoshop has taken layer management to its next logical evolutionary step with layer grouping (grouping layers by collapsible folders). If you haven't upgraded already, now is definitely the time.

Mac OS X Alert!

For those of you using OS X, rejoice! Macromedia has come through with the first release of a major non-3D graphics application for OS X, beating out their archrival, Adobe. Fully Carbonized, this application will run natively under both OS 9 and OS 10, meaning that you can take advantage of all the nifty bells and whistles of the new OS, fully integrated into one of the best illustration packages available (see Figure 2).

The first rule to remember about animation and design is that not every program can do everything, nor should it. Specialization can be a good thing, and a great time-saver. If it means that you get the job done sooner and make the client (and yourself) happier, then by all means, look for alternatives. Just as you would pick Photoshop over LightWave to do your image editing, the same applies to other peripheral aspects of your production pipeline. From the initial conceptual design work to the final compositing and output to your target media, you will find yourself building up a toolbox of applications to make your workflow more efficient and satisfying for you as an animator. A little wise investing now can save you many frustrating hours later down the road. Whereas I by no means advocate buying every new doodad that developers put on the shelf, keep your eyes open for new (or old) technology that can compliment your 3D application in a way that improves your work and makes you a dynamite sell to your clients.

Cailean Babcock has worked in computer animation for over five years, spending the last three at Educational Television Services at Oklahoma State University. He also moonlights in the wild and wacky world of freelance animation, with work featured on the Venture Outdoors series on Fox Sports, as well as his own character animation projects. His email address is cailean@provalue.net.

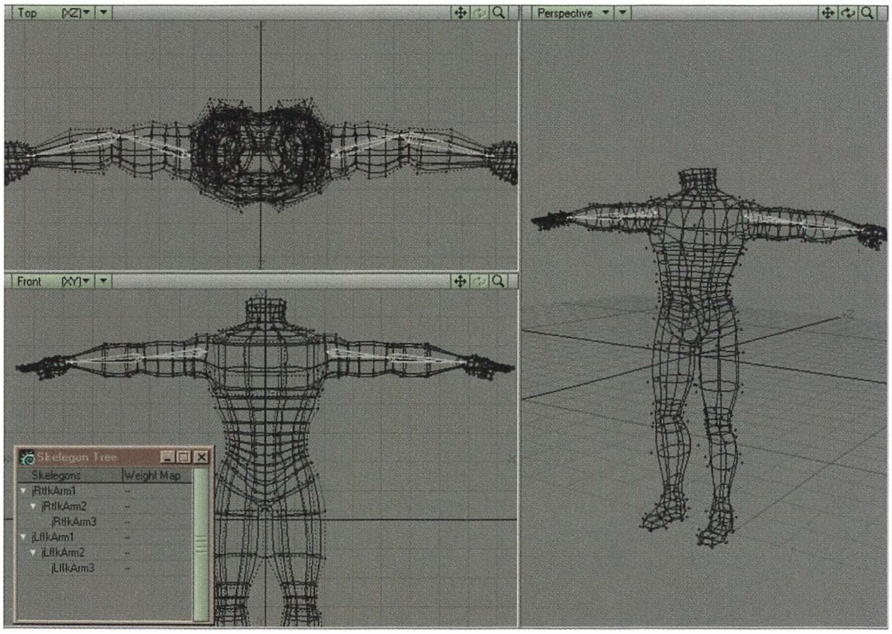
Advanced Character Setup Part II

by James Jacobs

In the previous issue, we rigged a character's hips, legs and feet. In this issue, we will cover the rigging of a character's torso and arms. Our attention will be focused on achieving attractive deformations, providing the animator with the flexibility of articulating their character's arms with either forward or inverse kinematics, (referred to from this point onward as FK and IK) and maintaining a streamlined user/puppet interface.

Before we can begin rigging our character, we must first study the motion of the joints that we wish to mimic.

Figure 1, above: *Spine Skelegons created and named.* ***Figure 2, below:*** *Arm IK Skelegons.*



1. The Spine
A person usually uses their spine in an additive manner. This means that the cumulative effect of each vertebrae's equal rotations accumulate along the length of the spine. This sort of motion is best handled by using a FK approach. Due to the number of joints that are involved in this movement, (seventeen bones starting at the base of the Lumbar region, and ending at the base of the neck, at the top of the Thoracic region) a hybrid approach must be taken in order to reduce the number of controls required to accurately articulate the character's spine.

2. The Rib Cage
Following Human Anatomy, we will parent the appropriate ribs to their corresponding vertebrae. We will also give the user control over chest expansion and contraction for the purpose of simulating breathing.

3. The Shoulder
The shoulder is the most complex joint structure in the human body, and thus serves as a challenge to character TDs. Due to the free floating nature of one's shoulder assembly, a combination of techniques must be employed in order to achieve realistic motion and deformation.

4. The Elbow
Loss of elbow volume when the joint articulates is a problem that plagues many character rigs. This rig will employ extra joints in the elbows to counteract this effect. This technique has the added benefit of creating the visible simulation of the humerus bone protruding from the elbow joint, when the joint is bent.

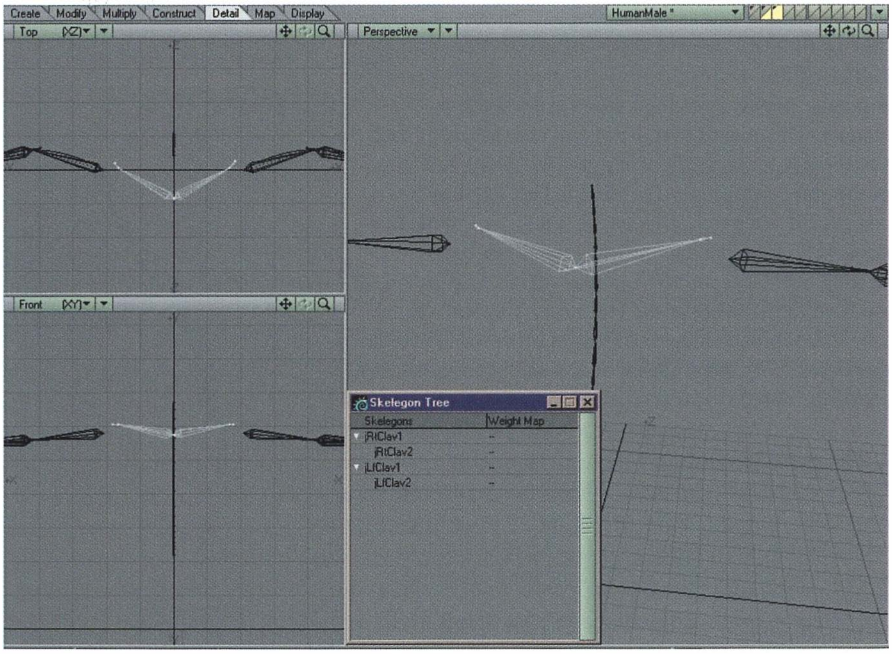
5. The Forearm
The forearm holds its own challenge. One must properly accommodate the twisting of the Radius and Ulna bones, as well as provide the appropriate degrees of freedom to the wrist. This rigging becomes further complicated when one wishes to give the added flexibility of a variable IK/FK solution, as we will be doing in this tutorial.

Step 1: Importing Geometry To Deform
The geometry that I will be using for the purpose of this tutorial is located in the following subdirectory to the LightWave directory: **Lightwave\Objects\Characters\HumanMale.lwo**

Step 2: Creating the Skelegons
Load the HumanMale.lwo object into your first layer. In the second layer, create your spine Skelegons, **Figure 1**. Name the Skelegons jSpine1 through to jSpine17.

In the third layer, create the Skelegons that will be used to provide the IK solution for the arms. Name them from jRtIkArm1 to jRtIkArm3 and jLfIkArm1 to jLfIkArm3 accordingly, **Figure 2**.

Figure 5: Clavicle skelegons.



Skelegons	Weight Map
jRtClav1	--
jRtClav2	--
jLfClav1	--
jLfClav2	--

Figure 6: Clavicle bones labeled.

Figure 4: Extra humerus bone.

Copy the contents of layer three, and paste it into layer four. Select jRtIkArm2 (while in layer four) and run Split Skelegon. Do the same with jLfIkArm2. Rename the joints jRtFkArm1 through to jRtFkArm4 and jLfFkArm1 to jLfFkArm4 accordingly, **Figure 3**. These Bones will be used for any FK animation that is applied to the arms.

Copy the Skelegons in layer four, and paste them into layer five. The bones in this layer will be the ones that will be used to deform the character mesh. This time rename the joints jRtArm1 through to jRtArm4 and jLfArm1 to jLfArm4.

Having finished renaming the arm Skelegons, create an extra bone that protrudes from the elbow on each side, **Figure 4**. Name these extra bones jRtHum and jLfHum. These Skelegons will both aid in retaining the volume of the elbow, as well as provide the visual simulation of the Humerus bone protruding from the elbow when the joint is bent.

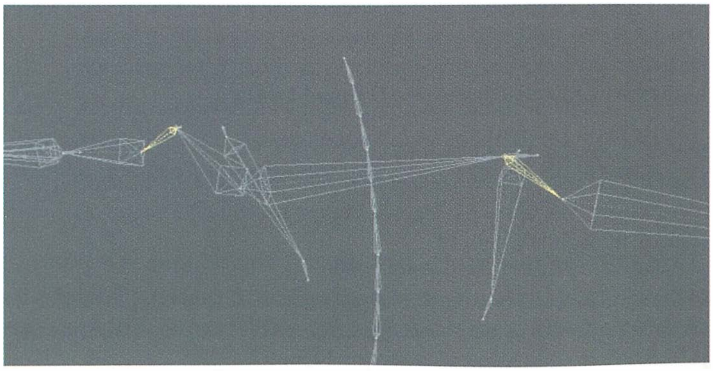


Figure 7: Shoulder joint bones.

Shift select layers three four and five, cut their contents and paste it into layer two.

In layer three, create the clavicle Skelegons, **Figure 5**. Label them jRtClav1, jRtClav2, jLfClav1 and jLfClav2 accordingly, **Figure 6**.Then in layer four, create the scapula Skelegons, and name them jRtScap1, jRtScap2, jRtScap3, jLfScap1, jLfScap2 and jLfScap3 accordingly.

In layer three, select the point that is shared between jRtClav1 and jRtClav2. Shift select the same point on the left side of the character. Copy, and paste the selection into layer two. Using these points, create two new bones labeled jLfShoulder and jRtShoulder, **Figure 7**.

Copy the clavicle Skelegons from layer three, and paste them into layer five. Move and scale the bones so that they can be used as ribs. Change the Skelegon names to jRtRib1, jRtRib2, jLfRib1 and jLfRib2 accordingly, **Figure 8**.

Cut layers three to five. Paste them into layer two. Add a Skelegon between jSpine14 and the top joints of the clavicle assembly. Name the new joint jClavicleGroup. Then do the same between jSpine10 and the top joints of the rib assembly. Name this new joint jRibGroup.

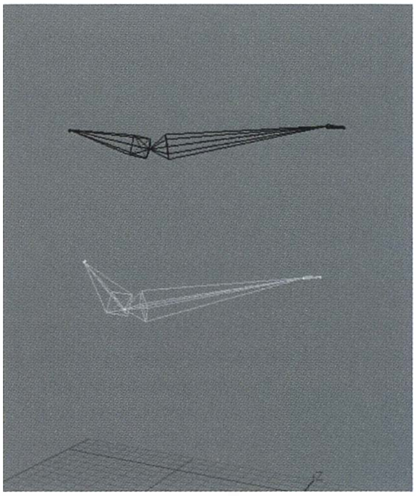
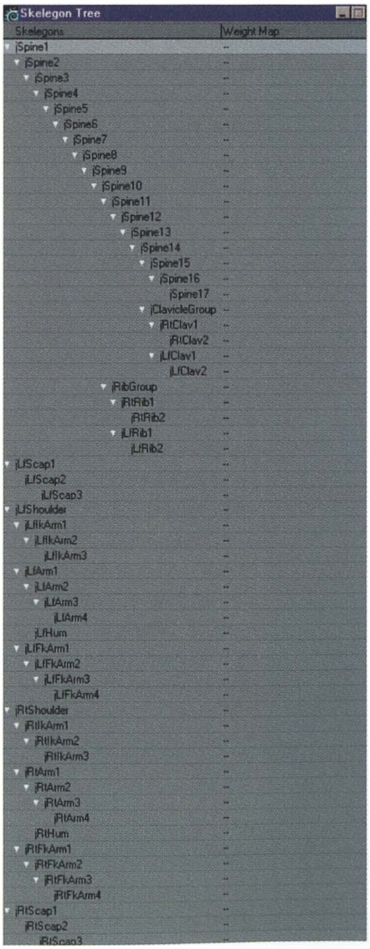


Figure 8: *Rib Skelegons.*



```
Skelegons                                 Weight Map
▾ jSpine1                                 --
  ▾ jSpine2                               --
    ▾ jSpine3                             --
      ▾ jSpine4                           --
        ▾ jSpine5                         --
          ▾ jSpine6                       --
            ▾ jSpine7                     --
              ▾ jSpine8                   --
                ▾ jSpine9                 --
                  ▾ jSpine10              --
                    ▾ jSpine11            --
                      ▾ jSpine12          --
                        ▾ jSpine13        --
                          ▾ jSpine14      --
                            ▾ jSpine15    --
                              ▾ jSpine16  --
                                  jSpine17 --
                            ▾ jClavicleGroup --
                              ▾ jRtClav1  --
                                  jRtClav2 --
                              ▾ jLfClav1  --
                                  jLfClav2 --
                    ▾ jRibGroup           --
                      ▾ jRtRib1           --
                          jRtRib2         --
                      ▾ jLfRib1           --
                          jLfRib2         --
▾ jLfScap1                                --
    jLfScap2                              --
      jLfScap3                            --
▾ jLfShoulder                             --
  ▾ jLfIkArm1                             --
    ▾ jLfIkArm2                           --
        jLfIkArm3                         --
  ▾ jLfArm1                               --
    ▾ jLfArm2                             --
      ▾ jLfArm3                           --
          jLfArm4                         --
      jLfHum                              --
  ▾ jLfFkArm1                             --
    ▾ jLfFkArm2                           --
      ▾ jLfFkArm3                         --
          jLfFkArm4                       --
▾ jRtShoulder                             --
  ▾ jRtIkArm1                             --
    ▾ jRtIkArm2                           --
        jRtIkArm3                         --
  ▾ jRtArm1                               --
    ▾ jRtArm2                             --
      ▾ jRtArm3                           --
          jRtArm4                         --
      jRtHum                              --
  ▾ jRtFkArm1                             --
    ▾ jRtFkArm2                           --
      ▾ jRtFkArm3                         --
          jRtFkArm4                       --
▾ jRtScap1                                --
    jRtScap2                              --
      jRtScap3                            --
```

Figure 9: *Complete Skelegon Tree.*

Figure 9 shows the completed Skelegon Tree. I've only displayed the left side of the character's hierarchy in order to conserve space. The right side is its mirror image.

Cut the contents of layer two and paste it into layer one. Save the model and send it into Layout.

Step 3: Preparing the Bones

Once in Layout, convert the model's Skelegons to Bones and arrange the Schematic view so that it clearly depicts the object hierarchy. Individually select each bone, and choose Record Pivot Rotation. The point of this is to zero out any rotations that are currently on the joints in order to simplify the process of expression writing. Select all of the Bones, then keyframe them in a rest pose at frame zero.

Step 4: Adding Spine Controls
Create five null objects. Label them SPINE1, SPINE2, SPINE3, SPINE4 and SPINE5. With Parent in Place mode off, group SPINE1 to jSpine2, SPINE2 to jSpine6, SPINE3 to jSpine9, SPINE4 to jSpine12 and SPINE5 to jSpine16. Toggle Parent in Place mode on. Parent SPINE1 to jSpine1, SPINE2 to jSpine5, SPINE3 to jSpine8, SPINE4 to jSpine11 and SPINE5 to jSpine15. Select each SPINE control and zero out its rotations, (set the value of each SPINE control's Heading, Pitch and Bank to zero) then keyframe them in a rest position at frame zero.

Step 5: Writing Spine Expressions
Each bone of our armature's back will be articulated by means of the five control nulls that we have created. In order to achieve a smooth interpolation along the length of the joints, expressions will be employed with the purpose to biasing the orientation of each joint to that of its one or two closest control nodes.

List 1, below, lists the bones in the character's spine, with their corresponding channel expressions.

jSpine2	Heading	SPINE1.rot(Time).x
	Pitch	SPINE1.rot(Time).y
	Bank	SPINE1.rot(Time).z
jSpine3	Heading	(SPINE1.rot(Time).x * .75) + (SPINE2.rot(Time).x * .25)
	Pitch	(SPINE1.rot(Time).y * .75) + (SPINE2.rot(Time).y * .25)
	Bank	(SPINE1.rot(Time).z * .75) + (SPINE2.rot(Time).z * .25)
jSpine4	Heading	(SPINE1.rot(Time).x * .5) + (SPINE2.rot(Time).x * .5)
	Pitch	(SPINE1.rot(Time).y * .5) + (SPINE2.rot(Time).y * .5)
	Bank	(SPINE1.rot(Time).z * .5) + (SPINE2.rot(Time).z * .5)
jSpine5	Heading	(SPINE1.rot(Time).x * .25) + (SPINE2.rot(Time).x * .75)
	Pitch	(SPINE1.rot(Time).y * .25) + (SPINE2.rot(Time).y * .75)
	Bank	(SPINE1.rot(Time).z * .25) + (SPINE2.rot(Time).z * .75)
jSpine6	Heading	SPINE2.rot(Time).x
	Pitch	SPINE2.rot(Time).y
	Bank	SPINE2.rot(Time).z
jSpine7	Heading	(SPINE2.rot(Time).x * .75) + (SPINE3.rot(Time).x * .25)
	Pitch	(SPINE2.rot(Time).y * .75) + (SPINE3.rot(Time).y * .25)
	Bank	(SPINE2.rot(Time).z * .75) + (SPINE3.rot(Time).z * .25)
jSpine8	Heading	(SPINE2.rot(Time).x * .25) + (SPINE3.rot(Time).x * .75)
	Pitch	(SPINE2.rot(Time).y * .25) + (SPINE3.rot(Time).y * .75)
	Bank	(SPINE2.rot(Time).z * .25) + (SPINE3.rot(Time).z * .75)
jSpine9	Heading	SPINE3.rot(Time).x
	Pitch	SPINE3.rot(Time).y
	Bank	SPINE3.rot(Time).z
jSpine10	Heading	(SPINE3.rot(Time).x * .75) + (SPINE4.rot(Time).x * .25)
	Pitch	(SPINE3.rot(Time).y * .75) + (SPINE4.rot(Time).y * .25)
	Bank	(SPINE3.rot(Time).z * .75) + (SPINE4.rot(Time).z * .25)
jSpine11	Heading	(SPINE3.rot(Time).x * .25) + (SPINE4.rot(Time).x * .75)
	Pitch	(SPINE3.rot(Time).y * .25) + (SPINE4.rot(Time).y * .75)
	Bank	(SPINE3.rot(Time).z * .25) + (SPINE4.rot(Time).z * .75)
jSpine12	Heading	SPINE4.rot(Time).x
	Pitch	SPINE4.rot(Time).y
	Bank	SPINE4.rot(Time).z
jSpine13	Heading	(SPINE4.rot(Time).x * .75) + (SPINE5.rot(Time).x * .25)
	Pitch	(SPINE4.rot(Time).y * .75) + (SPINE5.rot(Time).y * .25)
	Bank	(SPINE4.rot(Time).z * .75) + (SPINE5.rot(Time).z * .25)
jSpine14	Heading	(SPINE4.rot(Time).x * .5) + (SPINE5.rot(Time).x * .5)
	Pitch	(SPINE4.rot(Time).y * .5) + (SPINE5.rot(Time).y * .5)
	Bank	(SPINE4.rot(Time).z * .5) + (SPINE5.rot(Time).z * .5)
jSpine15	Heading	(SPINE4.rot(Time).x * .25) + (SPINE5.rot(Time).x * .75)
	Pitch	(SPINE4.rot(Time).y * .25) + (SPINE5.rot(Time).y * .75)
	Bank	(SPINE4.rot(Time).z * .25) + (SPINE5.rot(Time).z * .75)
jSpine16	Heading	SPINE5.rot(Time).x
	Pitch	SPINE5.rot(Time).y
	Bank	SPINE5.rot(Time).z

List 1: *Bones and channel expressions for the character's spine.*

Step 6: Creating the Shoulder Assembly
Create ten nulls and label them:

RT_SHOULDER
LF_SHOULDER
nRtShoulderTarget
nLfShoulderTarget
nRtScapulaSource
nLfScapulaSource
nRtScapulaTarget
nLfScapulaTarget
nRtRibTarget
nLfRibTarget

With Parent in Place mode off, group RT_SHOULDER and nRtShoulderTarget to jRtShoulder. Then group LF_SHOULDER and nLfShoulderTarget to jLfShoulder. This will move the affected null objects to the location of their new parent's pivot.

With Parent in Place mode on, unparent RT_SHOULDER and LF_SHOULDER. Zero their rotations and their Z position. Change LF_SHOULDER's X position to 7mm, and RT_SHOULDER's X position to -7mm. Parent each shoulder control to SPINE4. Zero the rotations out of the shoulder controls. Parent each shoulder target to their respective shoulder control, zero the target's rotations. This will move the shoulder control node's center of action closer to the character's spine, while keeping the shoulder's secondary pivot inside the shoulder's mass. Parenting the shoulder controls to the spine control, causes the shoulders to be affected by any animation that is placed on the back.

Turning Parent in Place mode off again, parent nRtScapulaSource to jRtScap1 and nLfScapulaSource to jLfScap1. Turn it on, and re-parent these nodes to nLfShoulderTarget and nRtShoulderTarget respectively. Then zero their rotations. These nodes will be used as constraint points for the parent bones in the scapula chains. Parenting nRtScapulaTarget to jRtScap3 and nLfScapulaTarget to jLfScap3 moves them into position as IK targets.

With Parent in Place mode reactivated, parent nRtScapulaTarget and nLfScapulaTarget to jSpine11.

Toggle the state of the Parent in Place option off yet again, and parent nRtRibTarget to jRtRib2 and nLfRibTarget to jLfRib2. Switch it on, and re-parent those nodes to jSpine10.

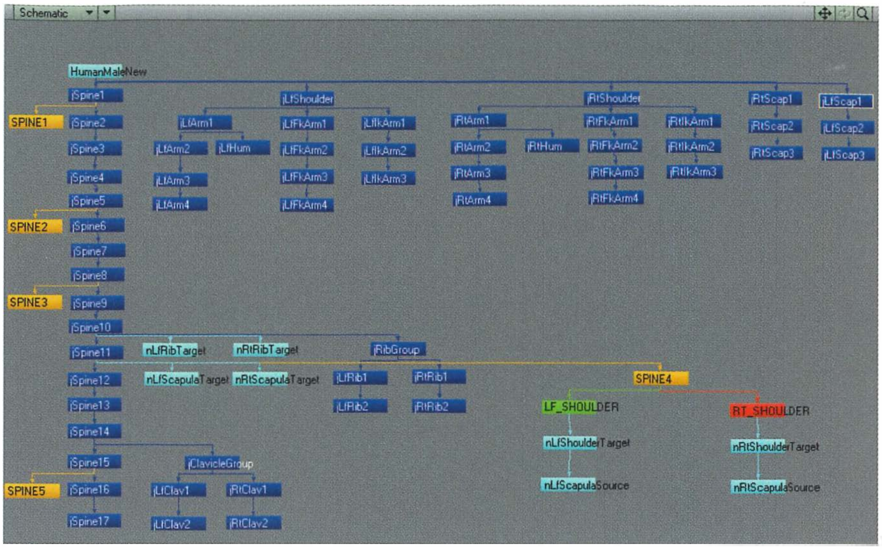

Schematic
HumanMaleNew
SPINE1
SPINE2
SPINE3
SPINE4
SPINE5
jSpine1
jSpine2
jSpine3
jSpine4
jSpine5
jSpine6
jSpine7
jSpine8
jSpine9
jSpine10
jSpine11
jSpine12
jSpine13
jSpine14
jSpine15
jSpine16
jSpine17
jLtArm1
jLtArm2
jLtHum
jLtArm3
jLtArm4
jLfShoulder
jLFkArm1
jLFkArm2
jLFkArm3
jLFkArm4
jLfIkArm1
jLfIkArm2
jLfIkArm3
jRtArm1
jRtArm2
jRtHum
jRtArm3
jRtArm4
jRtShoulder
jRtFkArm1
jRtFkArm2
jRtFkArm3
jRtFkArm4
jRtIkArm1
jRtIkArm2
jRtIkArm3
jRtScap1
jRtScap2
jRtScap3
jLfScap1
jLfScap2
jLfScap3
nLfRibTarget
nRtRibTarget
jRibGroup
nLfScapulaTarget
nRtScapulaTarget
jLfRib1
jLfRib2
jRtRib1
jRtRib2
LF_SHOULDER
nLfShoulderTarget
nLfScapulaSource
RT_SHOULDER
nRtShoulderTarget
nRtScapulaSource
jClavicleGroup
jLtClav1
jLtClav2
jRtClav1
jRtClav2

Figure 10: Hierarchy.

Your hierarchy should now resemble the one illustrated in **Figure 10**.

Step 7: Attaching the Scapula to the Shoulder Assembly
In order to isolate which rotations from the shoulder controls the scapula groups inherit, I make it my habit to have them constrained into the hierarchy, rather then directly parenting them. This is achieved through a combination of channel expressions and orientation constraints. **List 2** is a list of expressions applied to the bones in question.

Select jRtScap1 and open up its motion options. Choose as its Goal Object nRtScapulaSource. Turn on Match Goal Orientation and Full-Time IK. Change the rotations on the goal object so that the bone is oriented as it was before the IK option was activated. When nRtScapulaSource in rotated properly, set a keyframe at frame zero, and then record its pivot rotation. Follow the same procedure for the left scapula.

```
jRtScap1       PositionX       nRtScapulaSource.wpos(Time).x
               PositionY       nRtScapulaSource.wpos(Time).y
               PositionZ       nRtScapulaSource.wpos(Time).z
jLfScap1       PositionX       nLfScapulaSource.wpos(Time).x
               PositionY       nLfScapulaSource.wpos(Time).y
               PositionZ       nLfScapulaSource.wpos(Time).z
```

List 2: Expressions applied to the Scapula bones.

Step 8: Adding IK to the Clavicle, Scapula and Rib Assemblies
Select jLfClav1 and open up its motion options. Under the Controllers and Limits tab, choose Inverse Kinematics on the object's Heading, Pitch and Bank. Select jLfClav2 and in its motion options under IK and Modifiers, choose nLfShoulderTarget as its Goal Object, with Full-time IK on. Repeat this for the right clavicle.

Choose jRtScap2 and set its motion controller to IK on Heading Pitch and Bank. Set jRtScalp3's Goal Object to nRtScapulaTarget with Full-Time IK on. Do the same for the left scapula.

Select jLfRib1 and open up its motion dialog. Choose IK on the objects Heading and Pitch and Bank. Turn off all channels but Position X, Y and Z. Do the same for jRtRib1. Then set jLfRib2's IK goal object to nLfRibTarget and jRtRib2's goal to nRtRibTarget. Turn on Full-Time IK for both objects.

Apply the expressions shown in **List 3** to the appropriate bones.

Step 9: Attaching the Character's Arms
Select jLfShoulder and open it's motion options. Under Add modifier, choose Follower. Make jLfShoulder follow nLfShoulderTarget. Turn on Word Coordinates and After IK. Do the same for the other side of the character rig.

Step 10: Adding IK to the Appropriate set of Arms
Create two null objects. Label them LF_ARM and RT_ARM. With Parent in Place mode toggled off, parent LF_ARM to jLfIkArm3 and RT_ARM to jRtIkArm3. Turn on Parent in Place, un-parent the arm control nodes and zero their rotations.

On nodes jRtIkArm1, jRtIkArm2, jLfIkArm1 and jLfIkArm2 set their motion options in order to use IK on their Heading and Pitch. Then set jRtIkArm3 to Full-Time IK with RT_ARM as its goal and jLfIkArm3 to the same, only with LF_ARM as its goal.

Step 11: Add FK/IK Blending
We will use channel expressions to cause the deforming set of arms to choose which other set of arms it wishes to follow. First we must add an object for each arm, to act as an IK/FK blending switch. The Pitch of these new objects will bias the expressions on the arm joints to follow either the IK, or the FK arm. The range of the control is ninety degrees starting at zero and ending at positive ninety. To give our expressions channels to reference, we will add null objects which will serve as repositories of the IK joint rotational data.

Add six null objects named LF_IKFK, RT_IKFK, nLfIkArm1, nLfIkArm2, nRtIkArm1 and nRtIkArm2. Parent all of these new node to SPINE4, then add Motion Baker to the first two IK bones in each of the arms. Set Motion Baker to write out Extra Channels, and overwrite keys for the joint's Heading, Pitch and Bank.

After Motion Baker has been added to all of the appropriate bones, apply Channel Follower to the Heading, Pitch and Bank of the newly created null objects with the names that co-respond to the IK arm joints. Set channel follower to follow the AfterIK Heading, Pitch and Bank of the bones that have Motion Baker Applied. This step will allow the channel expressions that are on the deforming arm bones to see the joint rotations on those bones inside IK simulations.
Then add the expressions shown in **List 4** to the appropriate channels. (*Warning*: when switching from IK to FK or vice versa, the deforming bones may not update interactively without first scrubbing the time

```
jLfShoulder     Heading     (LF_SHOULDER.rot(Time).x >= 0)  *  (LF_SHOULDER.rot(Time).x)
                Pitch       (LF_SHOULDER.rot(Time).z >= 0)  *  (LF_SHOULDER.rot(Time).z)
jRtShoulder     Heading     (RT_SHOULDER.rot(Time).x >= 0)  *  ((RT_SHOULDER.rot(Time).x)  * -1)
                Pitch       (RT_SHOULDER.rot(Time).z <= 0)  *  ((RT_SHOULDER.rot(Time).z)  * -1)
```

List 3: Expressions applied to the Shoulder bones.

```
jRtArm1        Heading      ((nRtIkArm1.rot(Time).x) * (1 - (RT_IKFK.rot(Time).y /
                            90))) + (((jRtFkArm1.rot(Time).x) * (RT_IKFK.rot(Time).y / 90)))
               Pitch        ((nRtIkArm1.rot(Time).y) * (1 - (RT_IKFK.rot(Time).y /
                            90))) + (((jRtFkArm1.rot(Time).y) * (RT_IKFK.rot(Time).y / 90)))
               Bank         ((nRtIkArm1.rot(Time).z) * (1 - (RT_IKFK.rot(Time).y /
                            90))) + (((jRtFkArm1.rot(Time).z) * (RT_IKFK.rot(Time).y / 90)))
jRtArm2        Heading      ((nRtIkArm2.rot(Time).x) * (1 - (RT_IKFK.rot(Time).y /
                            90))) + (((jRtFkArm2.rot(Time).x) * (RT_IKFK.rot(Time).y / 90)))
               Pitch        ((nRtIkArm2.rot(Time).y) * (1 - (RT_IKFK.rot(Time).y /
                            90))) + (((jRtFkArm2.rot(Time).y) * (RT_IKFK.rot(Time).y / 90)))
               Bank         ((nRtIkArm2.rot(Time).z) * (1 - (RT_IKFK.rot(Time).y /
                            90))) + (((jRtFkArm2.rot(Time).z) * (RT_IKFK.rot(Time).y / 90)))
jRtArm3        Bank         jRtFkArm3.rot(Time).z
jLfArm1        Heading      ((nLfIkArm1.rot(Time).x) * (1 - (LF_IKFK.rot(Time).y /
                            90))) + (((jLfFkArm1.rot(Time).x) * (LF_IKFK.rot(Time).y / 90)))
               Pitch        ((nLfIkArm1.rot(Time).y) * (1 - (LF_IKFK.rot(Time).y /
                            90))) + (((jLfFkArm1.rot(Time).y) * (LF_IKFK.rot(Time).y / 90)))
               Bank         ((nLfIkArm1.rot(Time).z) * (1 - (LF_IKFK.rot(Time).y /
                            90))) + (((jLfFkArm1.rot(Time).z) * (LF_IKFK.rot(Time).y / 90)))
               Pitch        ((nLfIkArm2.rot(Time).y) * (1 - (LF_IKFK.rot(Time).y /
                            90))) + (((jLfFkArm2.rot(Time).y) * (LF_IKFK.rot(Time).y / 90)))
               Bank         ((nLfIkArm2.rot(Time).z) * (1 - (LF_IKFK.rot(Time).y /
                            90))) + (((jLfFkArm2.rot(Time).z) * (LF_IKFK.rot(Time).y / 90)))
jLfArm3        Bank         jLfFkArm3.rot(Time).z
```

List 4: *Expressions applied to the arm bones.*

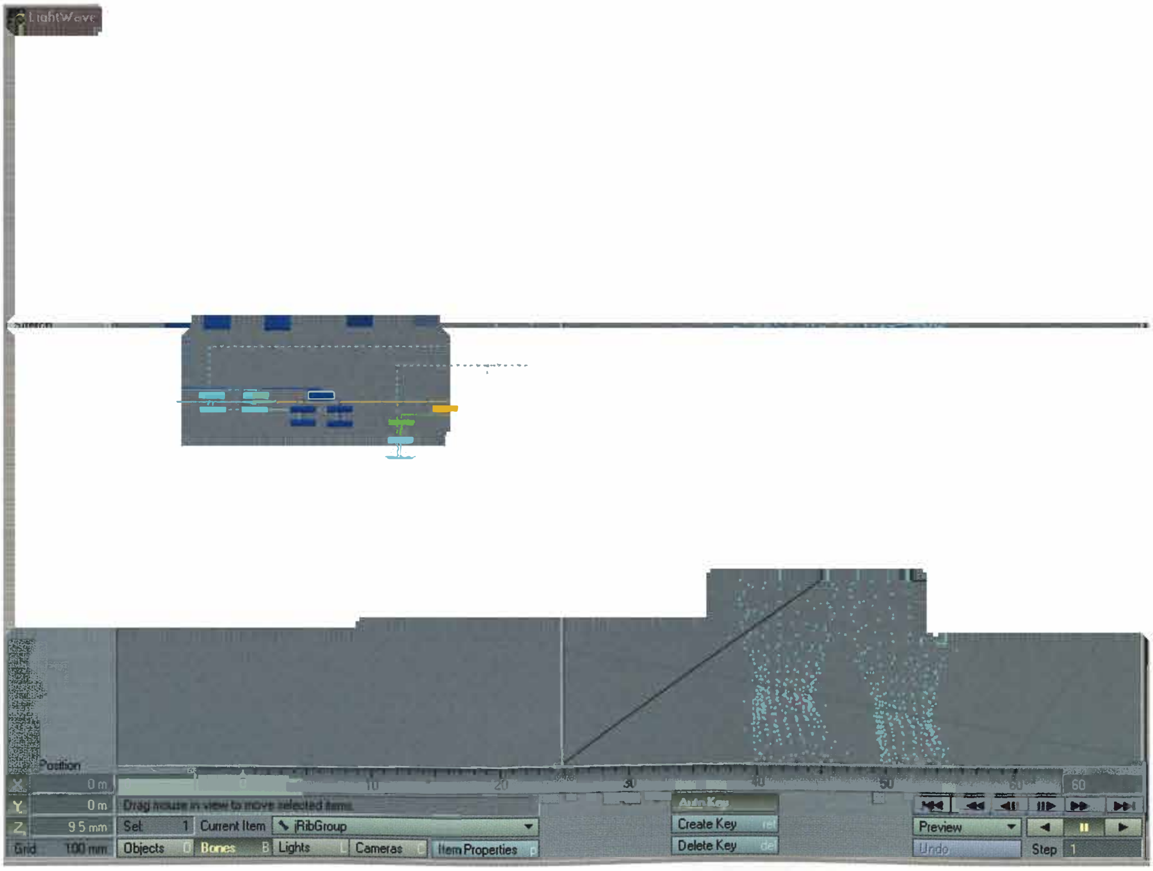
slider, and/or setting a keyframe.)

Step 12: Turn non Deforming Bones off, Lock Selection, etc. Select each bone in the arm IK and FK chains, and turn off its state. This will cause only those bones that are not directly articulated to effect the character mesh.

Lock the selection for all items except Control nodes, the bones in the FK arm chains and jRibGroup. Then turn off all channels that are not to be articulated by that particular control.

Figure 11 shows the final setup positioned inside the character's vertices.

If you wish to take this tutorial to its logical conclusion, you could replace the null objects with non rendering geometry, write expressions on the arm controls/bones to adjust their visibility depending upon the state of the IK blending controls, paint weight maps to refine the surface deformations, attach the arm twist bone to the IK goal and of course, add legs and a head.



Drag mouse in view to move selected items.

Figure 11: *Final setup with character geometry.*

I hope you have enjoyed this little sojourn into the world of character rigging. I look forward to traveling again with you in the future.

James Jacobs is a Senior Animator at Toybox, a visual effects and post production facility located in Toronto Canada. He has received an Emmy and a Gemini Award for his contribution to the production of Nelvana's Rolie Polie Olie, *and has been recognized in* Animation Magazine *for his work on a series of Hershey Kiss spots produced at Guru Animation Studio. James has filled such diverse roles as digital cel painter, compositor, render wrangler, technical director, overseas technical director (in Taiwan), senior technical director, and senior animator.*

The Professional Solution For Managing Network Rendering Has Arrived.

Along came a...

SPIDER™

- "Wanted to say thanks for all the help. **Spider** is running like a charm. Running **Spider** is like having the best personal assistant at your side coordinating your every rendering need. It's the only software to have if you're a luddite like me!"

  Kevin Kutchaver - President
  HimAnI Productions, Inc.
  www.himaniproductions.com

- "We have switched to using **Spider** for our network rendering. It works great. It even renders scenes that technically cannot be rendered (still scratching our heads on that one). Plus, there's more in store for the program down the road that look very promising. A reliable tool for network rendering... and not just with Lightwave. **Spider** is a must if you manage a large stack." "**Spider** rocks! You guys have a great product. I'm glad you guys are around or we'd be screwed."

  Micheal Hardison - CG Supervisor
  Pixel Magic
  www.pixelmagicfx.com

- "**Spider** is a great tool and very important in our pipeline. Without it we could not have finished our project "When Dinosaurs Roamed America" on time. **Spider** is very efficient, reliable and well designed for our production environment."

  Christophe Bernaud
  Director of Technology
  Meteor Studios
  www.meteorstudios.com

- "**Spider** helps us get more out of each machine on our network. No CPU's go to waste. I don't know what we'd do without **Spider**!"

  Sunny Wu
  Senior Network Administrator
  Shanghai Digital Film Studio
  www.sfs-cn.com

Spider graphic designed and modeled
by Dave Carlson and Pat Perez

Spider supports Windows 2000 and NT
Spider is TM and copyright © 2001 Spider Networks, Inc.

- "**Spider** really helps me manage my renders effectively when multiple project deadlines are fast approaching and every animator is clamoring to use the (render) farm. No matter how many processes I have, the render demand seems larger; the fact that **Spider** keeps the farm busy in an orderly and prioritized manner is a real asset." -RB

  "**Spider** takes our Lightwave and MAYA scenes and maximizes our render capabilities." -AP

  Rod Bland - Adrian van der Park
  VFX Animators - Rainmaker Digital Pictures
  www.rainmaker.com

- "We have been using **Spider** for over a year and Mitch has helped us a ton. Thanks again for the great package. The three full length cartoons we have done would not have made it without **Spider**!"

  ***Charles Meyer** - Wet Cement Productions*
  (Previously TRIAD Communications)
  www.triadcommunications.com

- "Since **Spider's** integration into our render farm, (50 workstations + 150 licenses of MAYA) managing our renders is simple and effective. **Spider**'s installation was easy and we appreciate the support **Spider** Networks provides."

  Alain Valois - Executive Director 3D Studio
  Mon Mannequin Virtuel Inc.
  www.myvirtualmodel.com

- "**Spider** has saved us endless hours of frustration. It was easy to set up and learn, and exceeded our expectations for ease of use. **Spider** has quickly become an indispensable part of our production pipeline. **Spider** is dependable and flexible, at an affordable price. Thank you! This is going to make our lives around here SO much better."

  Andrew Titcomb
  Senior Technical Director
  ABC Inc. Video Design Group

Website: www.stationx.com
Email: spider@stationx.com

Messiah Muscle, Part II

by Ron Wolfe

In Part I we looked at the various ways to create muscle motion in messiah. Of the different techniques, I pointed out that multiple skeletons offer a great deal of power and control. We're going to look at that technique and explore some ways it can be used to control muscles and skin.

To begin, let's breathe some life into Rex.

Making Rex breathe with multiple skeletons and expressions

Take a breath. Was that your normal, relaxed breath? Do you ever consciously think about breathing? Usually, not. It normally happens automatically in the background of conscious thinking and that's the way you want it. There is too much to think about during your life without concerning yourself with the physical task of breathing.

Our body changes the breathing rate depending on its activity level. When you are relaxed, your breathing rate is probably about 12 breaths per minute. If you become more active your breathing rate increases to adjust to the increased oxygen demand of your cells. Mowing the grass will increase your rate, but probably not to your maximum. Running will push your respiratory system so you breathe faster.

What does this have to do with making TRex breathe? We don't want to consciously control our own breathing, let alone that of our character. Make him do it the same way we do - automatically.

All you have to do is show him how to do it for one breath and tell him how often. Once you do that, Rex will keep breathing on his own and he'll do it "behind-the-scenes", so to speak, and not interfere with any other animation you want him to perform.

Breathing falls under the category of a "muscle motion" because the chest and abdominal muscles, the diaphragm and the airway get involved. We will train him how to use those muscles in his body. It takes some setup work to make Rex breathe, but once done you don't have to worry about it anymore.

The Breathing Rig

Load the scene named Rex_breathing.fxs into messiah. This is based on the TRex scene that comes with messiah, but with a breathing system added. The first thing you'll notice is that Rex himself is missing from the scene and there is a cube in his place. The Rex model comes with messiah and though I'm allowed to use him for examples, I can't just give him away. You will do an object replace to get your copy of Rex loaded.

Getting Rex loaded

1. Go to the Animate Tab, select Animation:Bone Deform and then click the Setup sub-tab to deactivate the current skeleton. Then do the same with the Breathing skeleton and finally the Muscles skeleton. All 3 skeletons should now be in Setup mode (**Figure 1**).

2. Go the Files Tab, Rep sub-tab. Select rex_cube.lwo and click on the Replace button. Browse to your folder that holds your copy of Rex. The file we are looking for is T_Rex_LowRes_01.lwo. My copy is in \messiah\messiah_content\Objects\T_Rex. When you find it, select the Open button to bring it into the scene in place of the Cube object.

3. Put the Muscles skeleton back into Keys mode first, and then the Breathing skeleton. Leave the Animation skeleton deactivated. Turn MetaNURBS on for Rex.

The Breathing Skeleton

The Breathing skeleton is not complex. It is a simple, functional skeleton designed to provide smooth deformation in the areas of the chest, abdomen, a small area of the neck, and the nostrils.

The Respiration Animation

For the respiration, I decided on a 96 frame looping animation. The actual breath occurs during the middle of this sequence and allows the sequence to loop. To see what it looks like, set your IN point to -300, your OUT point to -205, press Play and watch the chest area.

When you are done, set your OUT point to 1000 and your IN point to 0. Press Play again and notice that Rex keeps breathing. Now let's examine the expressions that allow him to use the animated breath and how they control its speed.

CycleBranchMorphAddFrame( )

Besides single frames, CBMaF can work on frame sequences. This makes it handy for getting Rex to breathe.

My choice of 96 frames for the respiration sequence was arbitrary. It works well because CBMaF stretches and shrinks the sequence according to the speed specified by the animator.

For instance, at 8 breaths per minute and 24 frames-per-second (FPS), there will be one respiration approximately every 7 seconds.

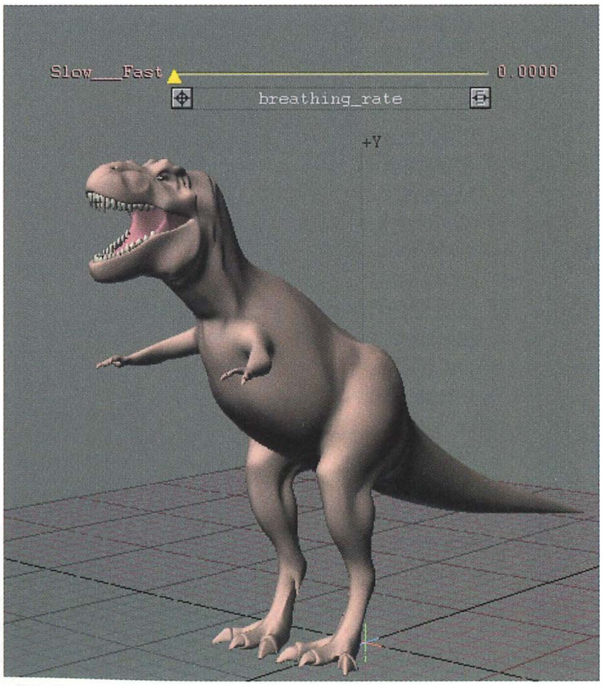


Figure 1

CBMaF takes the 96 frame sequence and stretches it to 168 frames (7 * 24). In contrast, if a rate of 24 breaths per minute is chosen, one respiration occurs every 2 seconds. CBMaF shrinks the 96 frame sequence down to 48 frames (2 * 24). You'll see that besides controlling the respirations, the expressions accommodate any FPS setting the animator uses.

Expressions for Automation

When building expressions you should think of specific goals. I came up with two for the breathing expressions:

- Require one simple input by the animator.
- Automatically compensate for the chosen breathing rate and the current frames-per-second setting.

That's it. Let's see what I came up with.

Expression #1:

```
Breathing_rate
floor * (60/ (([breathing_rate:xpos] * 16) + 8)))
```

Explanation: The slider that the animator uses to control the respiration rate is called "breathing_rate" (**Figure 1**). This expression processes that input. The inner part; (([breathing_rate:xpos] * 16) + 8))) remaps the values of the slider from 0.0 - 1.0 to 8.0 - 24.0. We want the minimum rate to be 8 breaths per minute and the maximum to be 24. If your character breathes within a different range, here is where you change the remapping values and the remaining expressions will adjust automatically.

The inner part retrieves the rate. If the breathing_rate slider is all the way to the left, the value will be 8.0. Once received, the rate is divided into 60 to determine how many seconds between respirations. 60/8 = 7.5. The "floor" part rounds down the float to an integer.

floor * (7.5) = 7

7 then is the number of seconds between breaths.

Expression #2:

```
Breathing_range
FPS * Breathing_rate
```

Explanation: FPS means frames-per-second and is a built in variable that reports the current setting.

This expression calculates the number of frames that are used for the respiration sequence, based on the Breathing_rate. I'm using messiah set at 24.0 FPS, so since we have already determined Breathing_rate equals 7, this expression calculates:

24.0 * 7 = 168

The Breathing_range is now 168 frames.

Expression #3:

```
B_factor
floor*(NOW/Breathing_range)
```

A Complex Production Requires Sophisticated Software for Managing Network Rendering

ButterflyNetRender™

"Without a doubt, I would say that ButterflyNetRender is the BEST distributed rendering solution. Not only does it support the most popular packages, but it's so easy to setup, and stable. Feature for feature, its leagues ahead of everyone else!"
– Todd Widup, animator, Dynamic Realities

Supports:
· LightWave 5.6 & LightWave 6.5
· Maya (Professional Ver.)
· Custom defined Jobs (Studio Ver.)

Features:
· Automatic system setup
· Frame Verification
· Scene Verification
· Real time RenderNode feedback
· Test Frame
· Email Notification system
· View Frame Directory
· Intuitive GUI +More features!

Professional Features:
· Group and Priority system
· Split Frames across RenderNodes

Standard version: $145.00
(includes 10 nodes)

Professional and Studio Versions competitively priced

awaits you at LiquidDreamSolutions.com

email: info@liquiddreamsolutions.com

Shipping and handling additional (plus sales tax where applicable)
(c)2001 Liquid Dream Solutions All rights reserved. ButterflyNetRender is a trademark of Liquid Dream Solutions. All other trademarks are properties of their respective owners.

Explanation: NOW is a timing variable in messiah and simply keeps track of the current frame. If you are on frame 154, NOW reports 154.

Here we are calculating a multiplier for the next expression. This will be used to repeat the breathing animation. The CBMaF function requires an input of 0.0 to 1.0.

From frame 0-167, we want the value to ramp between 0.0 and 1.0 for the first respiration. At frame 168, it needs to start over and ramp between 0.0 and 1.0 for the next 168 frames and repeat throughout the animation.

Let's say the current frame is 301. 301/168 equals 1.791666. The "floor" operator rounds this down to 1 and this is the value of B_factor.

Expression #4:

```
Breathing_loop
NOW - (Breathing_range * B_factor)
```

Explanation: Used to keep the reported frame between 0 and 167 throughout the animation. We're still at frame 301 and our B_factor is 1 so this expression calculates:

301 - (168 * 1) = 133

This is an offset of our respiration sequence and determines we are 133 frames (about 79%) into our Breathing_range of 168 frames.

Expression #5:

```
Breathing_increment
Clamp(Breathing_loop * (1.0/Breathing_range), 0.0,
1.0)
```

Explanation: We Clamp our output between 0.0 and 1.0. The part (1.0/Breathing_range) is defining the amount to increment the CBMaF function at each frame. We divide Breathing_range into 1.0 because 1.0 is the maximum value for CBMaF. We calculate 1.0/168 equals .005952. This is the amount incremented each frame.

Multiply this by Breathing_loop to calculate the value fed to CBMaF:

133 * (.005952) = .791616

This value tells CBMaF that we are about 79% (.791616) through the animation sequence for the respiration.

Expression #6:

```
Breathing_cycle_CBMaF
CycleBranchMorphAddFrame(breathing_branch,
breathing_branch, motion, -300+96 * Breathing_incre-
ment, 1)
```

Explanation: Here's our CBMaF expression and notice that we use Breathing_increment in place of a slider. You must understand what CBMaF is doing. We have defined the animation sequence for it to use starting at frame (-300) and continuing for 96 frames. The Breathing_increment value now directs it to a point in those 96 frames and it uses that point to position the Breathing bones at the current frame. It calculates 96 * Breathing_increment first and with a value of .791616 for Breathing_increment, this results in a value of 75.995136.

CBMaF now adds (-300) and 75.995136 to arrive at (-224.00487). It now knows that the current frame from our animation sequence is around frame (-224). With this set of six expressions Breathing_increment provides the value for CBMaF to use when stretching and shrinking the animation sequence and will repeat it for whatever length you need.

Rex Expressions Conclusion

Set the breathing_rate slider to different settings and see how the breathing changes in speed.

You can turn the Animation skeleton on now and see that Rex's breathing occurs beneath and does not interfere with the animation deformations.

To use the breathing_rate slider while animating, you keyframe it like normal. If Rex gets running somewhere in your scene and you want his breathing to increase, simply keyframe the slider at a higher value and Rex's breathing will automatically and smoothly increase. I hope you like the simplicity of this control.

Other Uses For Automatic Animation

Can we use the automatic technique for something other than breathing? Sure we can. It depends on your character, but for Rex we might also use it for blinking eyelids and tail wagging. You could even throw in some randomness. If a flying character, you sure wouldn't want to animate every flap of the wings. Maybe your character is drunk and he has the hiccups, or it's just something simple like a repeating walk cycle based on the distance the character has moved. An excellent example of this is on your messiah CD. Use it as a starting point to build a whole library of automatic animation for your character. The choices are up to you.

Before we end this article, I'm going to show you one more example of the multiple skeleton technique.

A Simple Example For Learning

Some years ago, I learned how to lay brick. I had built a few houses over the years and had always wanted to brick my own house someday. I was fortunate to have a friend who showed me the fine points of construction and one day he showed me how to brick. Before that day, it had always been some mysterious process to me and I didn't know where to start. But we mixed up the mortar, we had the proper trowels and tools and we had a good foundation to lay the bricks on.

My friend laid the first brick and I watched closely. Then it was my turn. Somewhat awkwardly I threw a line of mortar, buttered the end of the brick with some more mortar and set it beside the first brick. I struck the excess mortar from around my first brick and looked at my friend. He told me I knew how to lay brick now - just repeat it about 10,000 times and I could brick my house.

Why do I bring that up here? I was eager to build a brick wall, but I had to learn to lay that single brick first. That's what I'm going to do to show you this technique. I'll start with a very simple example (e.g. laying one brick) to direct you on your way to building more complex rigs. You must grasp the simple example first, because the complex rig is really not that hard. It is merely numerous repetitions of the simple one.

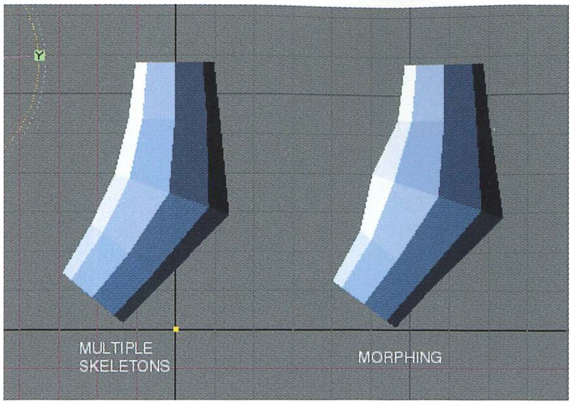


Figure 2

Comparing apples to apples

Load sample.fxs, press play and allow the scene to loop through frames 0-60. This is the simple object we are going to rig with multiple skeletons, but for now I wanted you to see what can be done. The three objects are all the same and you can see that the object tapers out at the joint. This type of joint presents difficulties when deforming because of the way it is built.

As it plays, look at the deformations of each of the three samples. Would you agree that the middle one seems to look the best? Well, it will probably be no surprise to you that the middle one is using two skeletons. The one to the right is using a morph and the other is plain bones. In defense of plain bones, keep in mind that messiah's bones are some of the best around but even they have no way of knowing the shape of the joint they're deforming.

Notice that the morph and multiple skeleton examples both look the same at the start and end frames, but the morph seems to act a bit weird in the middle (**Figure 2**). What's happening is the morph is being interpolated linearly between frame 0 and 60, and even though this means it will be 50% applied at frame 30, this just doesn't look right. It seems to widen and then reduce the size of the joint as the arm bends while the multiple skeleton example looks more natural. What is it doing that the morphing is not? It's easy.

While we're building our multiple skeleton we animate the muscle vertex_bones so the joint behaves the way we want. I hear the folks who like morphing screaming about multiple morph targets, and you're right - that would help, if you want to go to the trouble. But then you have to setup some more expressions to blend the multiple morphs and it may take more than one more morph target to work right.

What I'm telling you is multiple skeletons are simple and powerful. Let's set up a simple rig and show you how.

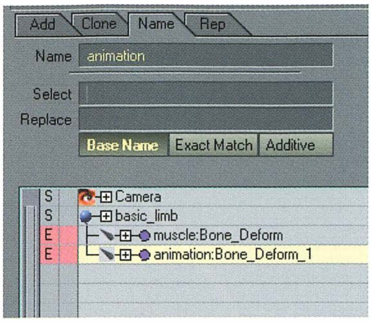
Applying The Technique: Starting Out

Clear messiah and load basic_limb.lwo. It's the same object we were just watching in the sample, but this time you are going to rig it.

Add the Bone_deform effects

Make sure basic_limb.lwo is selected and go to the Effect-Setup tab. Add a Bone_Deform effect, reselect basic_limb.lwo and add a second Bone_Deform effect (**Figure 3**).

Go to the File-Name tab and rename the first Bone_Deform to "muscle" and the second one to "animation". These are our skeletons.



Name: animation

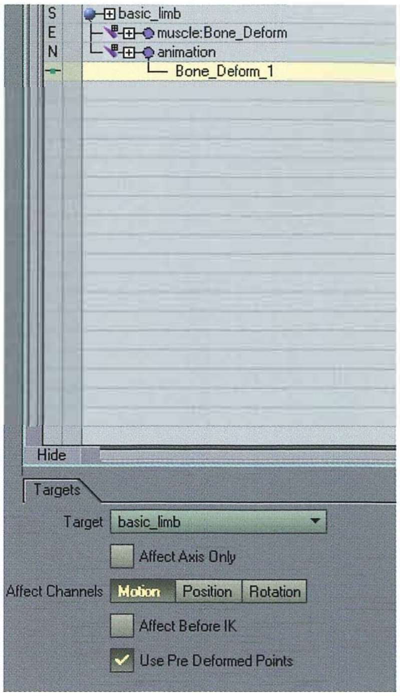
Select:

Replace:

```
S  Camera
S  basic_limb
E  ├ muscle:Bone_Deform
E  └ animation:Bone_Deform_1
```

Figure 3

Go to the Animate-Setup tab and click the little ball beside animation:Bone_Deform_1. Click on Bone_Deform_1 that drops down. In the Targets sub-panel, set the Target to "basic_limb" and check "Use Pre Deformed Points" (**Figure 4**). Click the little ball again to close.



```
S  basic_limb
E  ├ muscle:Bone_Deform
N  └ animation
       └ Bone_Deform_1
```

Target: basic_limb

☐ Affect Axis Only

Affect Channels: Motion | Position | Rotation

☐ Affect Before IK

☑ Use Pre Deformed Points

Figure 4

Adding bones to the animation skeleton

Select the animation skeleton and build a hierarchy of bones, like this:

animation:Bone_Deform_1
    UPPER_ARM
        elbow_ref
        FOREARM
            forearm_tip

The animation:Bone_Deform is at: 0,0,0 (XYZ) and 0,0,0 (HPB)

The bone positions and rotations for Setup are in **Figure 5**. Also set elbow_ref and forearm_tip to "Limited Range", Range = 0.0. "Z to Parent Length" for elbow_ref is unchecked.

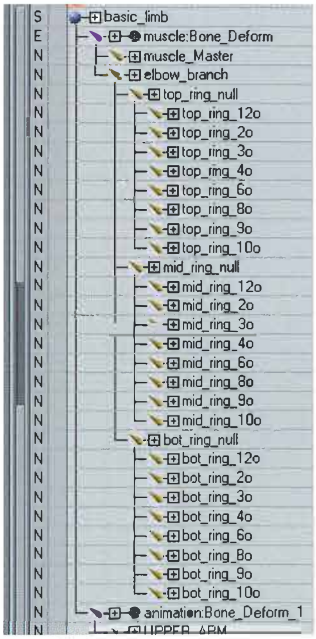
Adding bones to the muscle skeleton

Select the muscle skeleton. The muscle:Bone_Deform is also at 0,0,0 (XYZ) and 0,0,0 (HPB). The muscle skeleton hierarchy looks like this **Figure 6**.

Figure 7 (next page) shows the bone positions and rotations for the

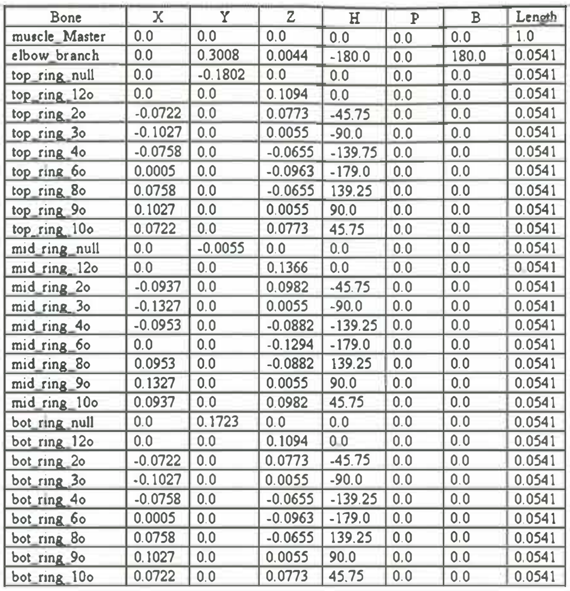
Bone	X	Y	Z	H	P	B	Length
UPPER_ARM	0.0	0.6111	0.0031	0.0	89.75	0.0	0.3086
elbow_ref	0.0	0.109	0.4316	0.0	0.0	0.0	0.0525
FOREARM	0.0	0.0	0.3086	0.0	1.5	0.0	0.3086
forearm_tip	0.0	0.0	0.3086	0.0	0.0	0.0	0.1

Figure 5



```
S  basic_limb
E  ├ muscle:Bone_Deform
N  │  ├ muscle_Master
N  │  └ elbow_branch
N  │     ├ top_ring_null
N  │     │  ├ top_ring_12o
N  │     │  ├ top_ring_2o
N  │     │  ├ top_ring_3o
N  │     │  ├ top_ring_4o
N  │     │  ├ top_ring_6o
N  │     │  ├ top_ring_8o
N  │     │  ├ top_ring_9o
N  │     │  └ top_ring_10o
N  │     ├ mid_ring_null
N  │     │  ├ mid_ring_12o
N  │     │  ├ mid_ring_2o
N  │     │  ├ mid_ring_3o
N  │     │  ├ mid_ring_4o
N  │     │  ├ mid_ring_6o
N  │     │  ├ mid_ring_8o
N  │     │  ├ mid_ring_9o
N  │     │  └ mid_ring_10o
N  │     └ bot_ring_null
N  │        ├ bot_ring_12o
N  │        ├ bot_ring_2o
N  │        ├ bot_ring_3o
N  │        ├ bot_ring_4o
N  │        ├ bot_ring_6o
N  │        ├ bot_ring_8o
N  │        ├ bot_ring_9o
N  │        └ bot_ring_10o
N  └ animation:Bone_Deform_1
N     └ UPPER_ARM
```

Figure 6



Bone	X	Y	Z	H	P	B	Length
muscle_Master	0.0	0.0	0.0	0.0	0.0	0.0	1.0
elbow_branch	0.0	0.3008	0.0044	-180.0	0.0	180.0	0.0541
top_ring_null	0.0	-0.1802	0.0	0.0	0.0	0.0	0.0541
top_ring_12o	0.0	0.0	0.1094	0.0	0.0	0.0	0.0541
top_ring_2o	-0.0722	0.0	0.0773	-45.75	0.0	0.0	0.0541
top_ring_3o	-0.1027	0.0	0.0055	-90.0	0.0	0.0	0.0541
top_ring_4o	-0.0758	0.0	-0.0655	-139.75	0.0	0.0	0.0541
top_ring_6o	0.0005	0.0	-0.0963	-179.0	0.0	0.0	0.0541
top_ring_8o	0.0758	0.0	-0.0655	139.25	0.0	0.0	0.0541
top_ring_9o	0.1027	0.0	0.0055	90.0	0.0	0.0	0.0541
top_ring_10o	0.0722	0.0	0.0773	45.75	0.0	0.0	0.0541
mid_ring_null	0.0	-0.0055	0.0	0.0	0.0	0.0	0.0541
mid_ring_12o	0.0	0.0	0.1366	0.0	0.0	0.0	0.0541
mid_ring_2o	-0.0937	0.0	0.0982	-45.75	0.0	0.0	0.0541
mid_ring_3o	-0.1327	0.0	0.0055	-90.0	0.0	0.0	0.0541
mid_ring_4o	-0.0953	0.0	-0.0882	-139.25	0.0	0.0	0.0541
mid_ring_6o	0.0	0.0	-0.1294	-179.0	0.0	0.0	0.0541
mid_ring_8o	0.0953	0.0	-0.0882	139.25	0.0	0.0	0.0541
mid_ring_9o	0.1327	0.0	0.0055	90.0	0.0	0.0	0.0541
mid_ring_10o	0.0937	0.0	0.0982	45.75	0.0	0.0	0.0541
bot_ring_null	0.0	0.1723	0.0	0.0	0.0	0.0	0.0541
bot_ring_12o	0.0	0.0	0.1094	0.0	0.0	0.0	0.0541
bot_ring_2o	-0.0722	0.0	0.0773	-45.75	0.0	0.0	0.0541
bot_ring_3o	-0.1027	0.0	0.0055	-90.0	0.0	0.0	0.0541
bot_ring_4o	-0.0758	0.0	-0.0655	-139.25	0.0	0.0	0.0541
bot_ring_6o	0.0005	0.0	-0.0963	-179.0	0.0	0.0	0.0541
bot_ring_8o	0.0758	0.0	-0.0655	139.25	0.0	0.0	0.0541
bot_ring_9o	0.1027	0.0	0.0055	90.0	0.0	0.0	0.0541
bot_ring_10o	0.0722	0.0	0.0773	45.75	0.0	0.0	0.0541

Figure 7: .

muscle skeleton. A special note as you build the skeleton: The elbow_branch, top_ring_null, mid_ring_null and bot_ring_null bones are all inactive, so set them to Limited Range of 0.0 as you build them. ALL the bones that end with 'o' have a Limited Range of 0.02. Every bone in the muscle skeleton has "Z to Parent Length" unchecked:

Okay, did you add all those bones? Well, don't get mad but now I'm going to tell you that if you had trouble or decided not to do it, you can load the file called basic_limb_b.fxs and its all done for you. Ensure that both skeletons are in Keys mode.

For those of you who did add all those bones manually - good job. You've gotten a little bit of the mortar on your hands now, and that helps you learn.

Establishing our Plane

Using the AngleBetweenTri( ) function: Though we could just use the actual bone rotation values, I want you to learn this function because it is an important part of the technique. With it, the technique works with or without IK, avoids the flipping of axis when using local coords and does not have to be setup on any specific channel.

Go to the Command-Expression tab and click "New Expression". Name it "Elbow_Angle" and enter the following in the Exp box:

```
AngleBetweenTri(FOREARM,elbow_ref,forearm_tip,NOW,NOW,
NOW)
```

Turn "debug" ON for the Command panel. Now select the FOREARM bone and rotate it on world pitch (left mouse button). Watch the changing values for the expression as you do this. It's working fine but the reading is in radians. I like to work in degrees so let's change the expression by adding RTOD:

```
RTOD*AngleBetweenTri(FOREARM,elbow_ref,forearm_tip,NOW
,NOW,NOW)
```

Now the value is in degrees and I find this easier. We need to determine the range of this joint. Rotate the FOREARM bone again and watch the value on the Command panel. I came up with a range of 43.0 to 135.0 degrees. Next, clamp our expression within these values:

```
Clamp(RTOD * AngleBetweenTri(FOREARM,elbow_ref,
forearm_tip,NOW,NOW,NOW),43,135)
```

All the expressions calculate after IK. Okay, that expression's done. Now we're going to use the angle to determine how we apply the muscle skeleton bones. Add the following expression:

```
Elbow_Percentage
(Elbow_Angle - 43) /92
```

Rotate the FOREARM again and watch the percentage reading. You'll notice it's around 0.0 when the arm is straight and at 1.0 when we bend it up to 135 degrees. This is exactly what we want - a value that increments from 0.0 to 1.0 while the joint is rotated. Before we add the final expression, we have to do some animation.

Animating the Skin with the muscle skeleton

There are two ways to use the CycleBranchMorphAddFrame( ) function for our needs. One is for it to use the position of bones at a single frame and the other is for it to use the animation of bones over a sequence of frames. The sequence can be any length and placed anywhere but it is best to put it in a negative range. I have decided on frame (-200) to frame (-101).

Go to frame -200 and key the FOREARM bone so it is at its default, straight position. These rotation values are:

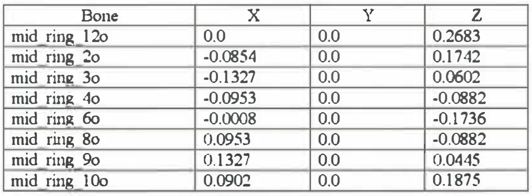
0.0, 1.5, 0.0 (HPB)

Do the same for frame -99.

Now, at frames -101 and -100 key the FOREARM to this:

0.0, 94.25, 0.0 (HPB)

Now scrub through frames -200 to -101 and the arm bends. Next, we use the muscle vertex_bones to enhance the deformation at frame -101. Select the elbow_branch bone and go to frame -200. Click the animation window to make it active, then hit ENTER, and pick "Current Item & Descendants" and click OK. This makes a key for every single bone under elbow_branch. Do the same thing at frames -101, -100 and -99. Now go to frame -101 and we'll work with the joint. We'll use the bones under the mid_ring_null bone first, because they make the biggest difference. So at frame -101, key the following XYZ changes (**Figure 8**). Make certain the key at -101 is selected. It is normal that the muscle vertex_bones are not aligned with the geometry.



Bone	X	Y	Z
mid_ring_12o	0.0	0.0	0.2683
mid_ring_2o	-0.0854	0.0	0.1742
mid_ring_3o	-0.1327	0.0	0.0602
mid_ring_4o	-0.0953	0.0	-0.0882
mid_ring_6o	-0.0008	0.0	-0.1736
mid_ring_8o	0.0953	0.0	-0.0882
mid_ring_9o	0.1327	0.0	0.0445
mid_ring_10o	0.0902	0.0	0.1875

Figure 8

That makes it look quite a bit better, doesn't it? I have to watch my space here, and since their changes are almost negligible, I'm going to skip top_ring_null and bot_ring_null right now, but you'll be loading an updated scene shortly so don't worry about it. I want to show you the last little bit that makes this technique superior to others.

Finding and correcting anomalies

Scrub through frames -200 and -101 and you should see the arm

Continued on Page 28

ARE

Foundatio
Institute
Classes and

Tel. 661·257·0292

Fax 661·257·7966

www.foundation-i.com

24933 W. Avenue Standford

Valencia, CA 91355

Bone	X	Y	Z
mid_ring_12o	0.0	0.0	0.1722
mid_ring_2o			
mid_ring_3o			
mid_ring_4o	-0.105	0.0	-0.0686
mid_ring_6o	-0.0001	0.0	-0.1327
mid_ring_8o	0.0933	0.0	-0.0771
mid_ring_9o	0.1327	0.0	0.0275
mid_ring_10o	0.0905	0.0	0.1331

Figure 9

bend and the muscle bones adjusting the skin at the joint. Do you notice that, right now, this looks exactly like the morphing example we saw earlier? That's because the muscle bones are being interpolated through the 100-frame sequence. I think it looks bad at frame -150. Go there now and notice how the joint looks like it is too wide for the current rotation. Well, guess what? We're going to fix it right here. Using the same bones again, make a key at frame -150 with the following values in (**Figure 9**).

The bones with blank fields do not need this key at frame -150, so only change the ones indicated. Now play through the sequence. Looks a lot better, doesn't it? One more thing, and then we're done. First load basic_limb_c.fxs so we're all working with the same file. You'll notice that the top_ring_null and bot_ring_null have just a small amount of animation here, but like I said - it's negligible. Let's finish up this joint by making the enhancements work whenever the FOREARM is rotated - completely automatic.

Go to frame 0. Set your IN point to 0 and OUT point to 60. Key the FOREARM bone at 0 with 0, 1.5, 0 (HPB). Key it at frame 60 with 0, 94.75, 0 (HPB). Play through this new range and there it is again, just rotating. But, as you've probably noticed, the muscle bones are not doing anything - yet. For them to work, we need the CBMaF function, so go to the Command-Expression tab and add the following expression:

```
Elbow_CBMaF
CycleBranchMorphAddFrame(elbow_branch,elbow_branch,mot
ion,-200 + 99 * Elbow_Percentage, 1)
```

Set it to work after IK and you're done. Now replay that range and notice the improvement. You can rotate the FOREARM bone at any time and the skin will automatically adhere to the form you animated in those negative 100 frames.

The CBMaF function uses "99 * Elbow_Percentage" to calculate what frame to reference in positioning the bones. If our FOREARM bone is rotated 60% (.60) through its range, this will calculate out to 59.4. That is then added to -200 to arrive at frame -140.6 and that's what CBMaF uses. It recalculates this constantly whenever the FOREARM rotates!

Wrap up and editing tips

If you had any problems, just load basic_limb_d.fxs to look at the completed scene. You can edit the sequence from -200 to -101 at any time, and the changes will update automatically throughout your animation. One warning though, always remember to turn OFF the CBMaF expression before editing or else the movement of the muscle vertex_bones will be compounded. This will make more sense if you ever forget to do it.

Well, that was the single brick version. Now, I'll let you look at this technique replicated multiple times in a more complex rig. I've included a file called Stego_rig.fxs in the files (**Figure 10**). I could only let you have a partial object, but most of the action is in the legs so it is still a good example. It uses the technique on the knee joint, ankle joint, calf muscle and hamstring and quadriceps muscles of the legs. Study it at your leisure and remember, I built it one brick at a time and so can you.

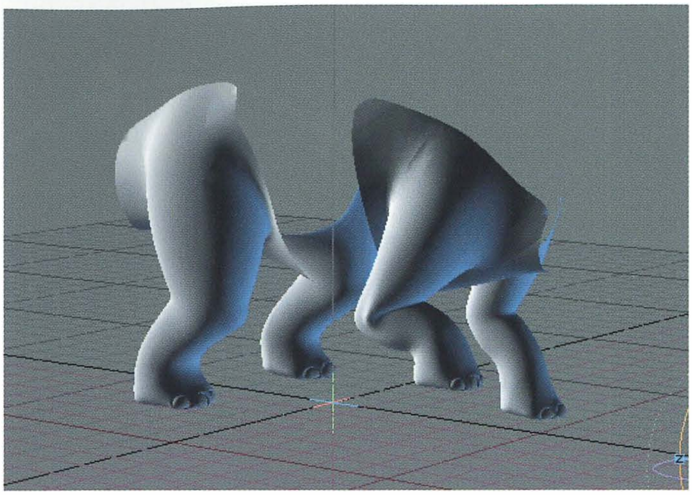


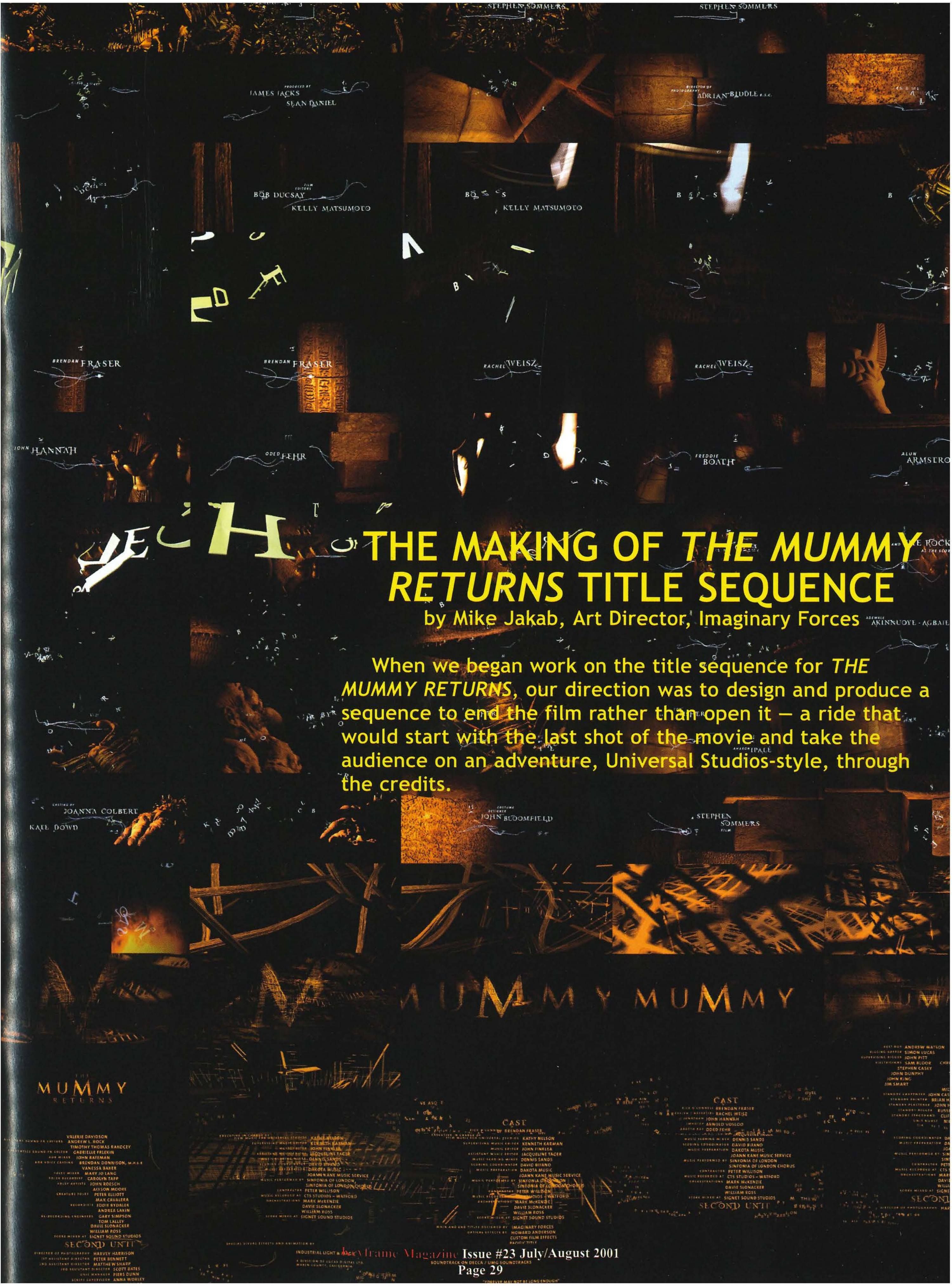
Figure 10

Conclusion

You've seen a variety of ways to make muscles move. Remember this article was designed to show you another way to move muscle and skin. It's not the only way, but the method I prefer. If you're comfortable with morphs - use those. If you like to stick with puppet_master - use that. But, if you ever run into that rigging dead-end where you're not getting the results you want, give this method a try. I think you'll find it worth the effort.

My thanks to pmG for their support and the use of the TRex model.

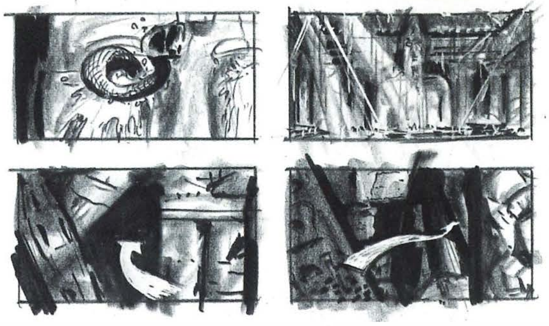
Ron Wolfe is a free-lance animator and messiah enthusiast who works out of his home on various projects. He works via the Internet and has recently been animating 3D characters for games with messiah. If you have any questions or comments, please feel free to drop him an email at rwolfejr@woh.rr.com.

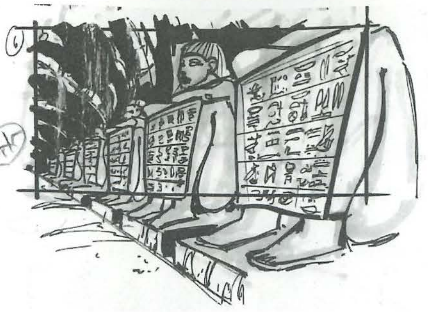

STEPHEN SOMMERS
PRODUCED BY JAMES JACKS
SEAN DANIEL
DIRECTOR OF PHOTOGRAPHY ADRIAN BIDDLE B.S.C.
FILM EDITORS BOB DUCSAY
KELLY MATSUMOTO
BRENDAN FRASER
RACHEL WEISZ
JOHN HANNAH
ODED FEHR
FREDDIE BOATH
ALUN ARMSTRO
ADEWALE AKINNUOYE-AGBAJE
CASTING BY JOANNA COLBERT
KATE DOWD
COSTUME DESIGNER JOHN BLOOMFIELD
STEPHEN SOMMERS
MUMMY MUMMY
THE MUMMY RETURNS
CAST
SECOND UNIT

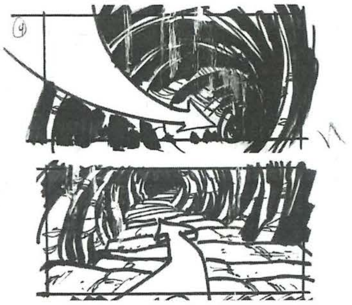
THE MAKING OF *THE MUMMY RETURNS* TITLE SEQUENCE

by Mike Jakab, Art Director, Imaginary Forces

When we began work on the title sequence for *THE MUMMY RETURNS*, our direction was to design and produce a sequence to end the film rather than open it — a ride that would start with the last shot of the movie and take the audience on an adventure, Universal Studios-style, through the credits.







The first step in our design phase was to read the script and develop storyboards that explored different conceptual approaches to the sequence; our ideas were influenced by the elements of the story: gold, sand, etc. We presented these ideas to the director; after our first meeting, he asked us to combine two of the concepts into one for the ultimate design — a dark journey through the catacombs and brick tunnels where type is constantly forming in the distance as you travel to your uncertain destination.

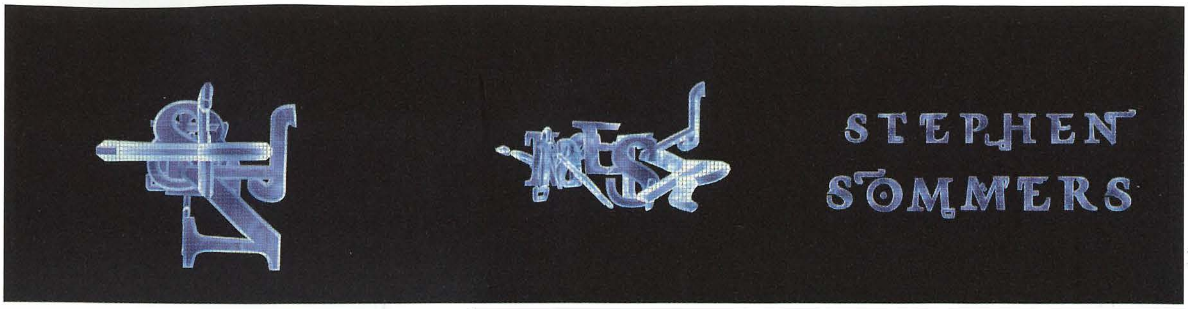
Because we were not going to be able to go shoot the catacombs in Egypt, we had to develop another methodology to produce the sequence. We started with production stills. With access to all of the MUMMY images from Universal Studios, we selected the ones we needed and got them in digital form to use as reference.

Without actual footage, we knew we were going to have to create the catacombs in 3D. In our initial tests, we created a 3D dirt tunnel that was dimly lit and animated in such a way that the viewer would never know where they were going; in these tests, we started to introduce bricks rendered in 3D with MAXON's Cinema 4D software.

We did not know that much about the program yet, but based on the success of these initial experiments with the software, we quickly saw the possibilities of our sequence to expand as we discovered how realistic the textures and movements could be when created and rendered in Cinema 4D.

Creatively we knew it was going to work, but there were still technical questions. We needed to work at 2K resolution, for output to 35mm film, and did not know if Cinema 4D would be compatible. All of our testing proved that it would be; we determined the appropriate file sizes and the render times.

With all of our research now complete, we began creating our store of elements. Using a stock ground and rock texture we bought online and processed through Photoshop, we designed 25 hieroglyphic walls that ranged from simple to complex. These walls were intended to always be a part of the sequence and never to stand-alone. This set-up gave us the freedom to focus on the details and the form of the wall with elements like flame flicker and atmospheric fog.



With the design of the backplate almost locked, we began working on how we would integrate the twenty-five credits, and the main title card. We explored two directions for the animation of each credit. One was based on having a hieroglyphic-like form scale up, rotate, and evolve into the type. It was an interesting idea, but not executed as well as the concept that we ultimately chose.

Since we were traveling through the tunnels, it seemed natural for the typography to fly around as if trapped by a stream or current of air. In this second motion test, individual letters would fly together to form a credit and blow apart to bring on the next credit. The idea worked perfectly, and it had just enough variety to keep the motion exciting for all twenty-four cards.

To keep the overall look consistent and dimensional, we were going to use Cinema 4D to animate all of the type, but first we had to set the legal projections in Illustrator to establish the legal size and type style. We placed each card on a field chart, in this case it was 2.35:1, for the studio's approval of size and placement. We explored many variations and type designs before we came up with the one that was ultimately approved. In the final set, we added the ornate, thin lines representative of gauze that would be embedded in the cards as they landed in frame. These lines would, piece by piece, create our end logo.

We experimented with different lighting techniques and letter depths on our custom typeface and decided to keep it white with a slight extrusion. It was a traditional approach to white type on a black background but one that would now incorporate depth and spatial relationships.

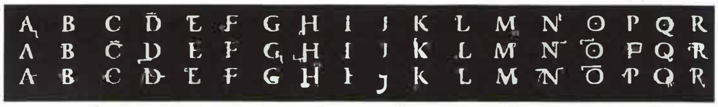
Even with everything we had created, we still needed some live action elements. We had a Mini-DV shoot with out Canon XL-1. Billowing gauze was our main attraction. After we digitized the footage, we incorporated it into our animated type sequence and 3D elements.

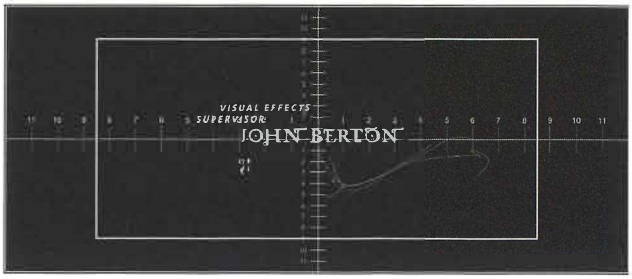
When we showed our first cut to the director, he said: "We love it, but it's a bit too dark and scary." He stressed that The Mummy Returns was going to be more of a family/action/romance film and that we should keep that in mind while we were revising the sequence.

Based on the director's notes, we went back into the Avid and lightened the sequence up by losing a few of the skeletons and brightening up some of the bricks. We presented to the director again. While he still had changes, they were only mild timing issues.

Now it was time to re-build the entire video resolution sequence in the Inferno at film resolution. All of the files that were originally created at 720 x 486 now had to be recreated at 2048 x 1556, or 2k; we needed serious memory — half a terabyte — for the all of the rendering that was about to take place.

When we began the project, we had underestimated the learning curve on C4D so that by the end we had improved our animations and renders a great deal. But this meant that we had old models and textures mixed with new ones. For our final composite, we decided to make all the older animations as amped up as the new ones. This added to the already massive number of 2K files we were re-rendering. This was serious; we brought in Cinema 4D Net so that one computer on our server would be linked to all of the other processors and we could render with great speed. At one time, we had a total of 43 computers rendering final CGI sequences for film. Quite impressive when you aren't a render house expected to pack that kind of power.


A B C D E F G H I J K L M N O P Q R
A B C D E F G H I J K L M N O P Q R
A B C D E F G H I J K L M N O P Q R



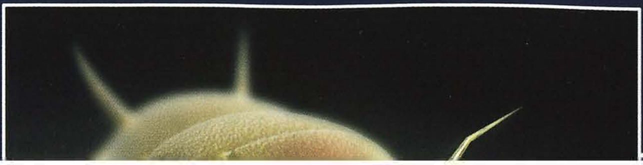
While the files were rendering, we set up a full day, hi-def shoot of props at the Universal Archive to get the hi-res elements we needed for the final sequence. We shot many of the actual props used in the film — swords, gold bracelets, skeletons, etc. — and we went to where the sculptors were making the gold covered eight-foot high statues and other sculpted props. We included two live scorpions in the shoot and used their scenes in the backplate for one of the main actor's cards.

As everything was being finalized in the Inferno, we went back to refining some small details, including the actual The Mummy Returns logo. The twenty-four cards that led to the logo were animating on over the backplate, but the main logo didn't have a special treatment. We needed to make the thin line of gauze that was a part of each credit mean something. Using Cinema 4D's capabilities with animated alphas in the textures and bent geometry, we eventually built an illusion of growing, wrapping gauze. The final title animation then fit within the entire graphic language of the piece.

We soon learned that rendering out each layer of the main logo into separate passes would work much faster; with separate layers our Inferno artists could easily adjust timings and have more flexibility: the gauze could wrap faster, or slower, or scale differently.

Finally, there was the endcrawl. We had to seamlessly go from the main logo into the crawl of names that would change up until the very last minute before delivery. To produce the end sequence and pieces of the crawl separately is not a problem, except for the issue of non-matching blacks; when the two pieces of negative are cut together, they will never be the same because they were shot at different times. We had to come up with a break in the sequence that would divert the eye from even the possibility of non-matching blacks. We created these breaks with our 3D gauze elements and planned on having three within the crawl. In the end, however, we didn't use these cut-points, and we finished the entire sequence as one piece of negative. It was a marvel: one complete digital end crawl. The film-out time alone was 14 hours.

So that is how we did it — from beginning to end. We worked countless hours, put our lives on hold for a bit, gave it everything, and pulled off something pretty amazing in relatively no time. It goes without saying that there was an incredible team of people who contributed to this project. Without all of their dedication, support, and knowledge, this exciting ride would never have been possible.



CINEMA

3D Modeling,
Animation,
Rendering

"Memories" by Axel Ritter

Available for
Macintosh | Windows

"Cat & Bear" by Joachim Heller

BODYPAINT 3D

3D Painting,
Texturing,
UVW Editing

"Egg Chamber" by Jeff Carns

"Caution, may cause inspiration..."

Free Demo Versions
www.maxoncomputer.com

SIGGRAPH Booth #1430

Diving in with Caustics and Deformation Modeling with bhodiNUT's NICKL

by Christopher Mills

What are Caustics? Not a harsh chemical, they are actually a class of mathematical expressions used to describe the interaction of light reflecting off of and refracting through different optically transmissive materials. A classical description of Caustics reads:

"Caustic curves: When light reflects off a curve then the envelope of the reflected rays is a caustic by reflection or a catacaustic. When light is refracted by a curve then the envelope of the refracted rays is a caustic by refraction or a diacaustic. They were first studied by Huygens and Tschirnhaus around 1678. Johann Bernoulli, Jacob Bernoulli, de l'Hôpital and Lagrange all studied caustic curves."[1]

In more practical terms, Caustics are found pretty much anywhere that you find metals, liquids and light. The pattern of light found at the bottom of a swimming pool filled with water is an expression of caustics; the way a magnifying glass condenses and focuses the rays of the sun to set fire to a piece of paper is also an expression of caustics. The dazzling pattern of light reflected from a chrome sculpture is also an expression of caustics in the real world.

The use of caustics in a 3D rendered scene can lend a degree of realism to the finished results without costing too much in terms of rendering time. Through careful use of the caustic settings in CINEMA 4D XL v7, you can achieve a high degree of photo-realism in your finished renders. The goal of this tutorial is to show you how to control the various parameters for both surface and volumetric caustics. One of the main points will be to create believable looking water - both the surface and caustics visual effects associated with water. A great way to do this is with the new deformer tool from bhodiNUT called NICKL. NICKL is free to any registered user of C4D XL from MAXON's Plug-in Café.

NICKL includes several noise-based functions for use as within a deformation tool in C4D XL. Noise functions can add a heightened degree of realism to 3D images and can be applied to nearly every aspect of scene or animation. Used in modeling, texturing, lighting or animation; noise-based functions can add an organic quality that would otherwise be very difficult to produce by hand. Ken Perlin is credited with the creation of the algorithms that are now used to produce most of the noise functions used in computer graphics from the 1980s forward. The caption on his Technical Achievement Award from the Academy of Motion Picture Arts and Sciences for work he did back in the 1980s on procedural textures sums it all up with:

"To Ken Perlin for the development of Perlin Noise, a technique used to produce natural appearing textures on computer generated surfaces for motion picture visual effects. The development of Perlin Noise has allowed computer graphics artists to better represent the complexity of natural phenomena in visual effects for the motion picture industry."[2]

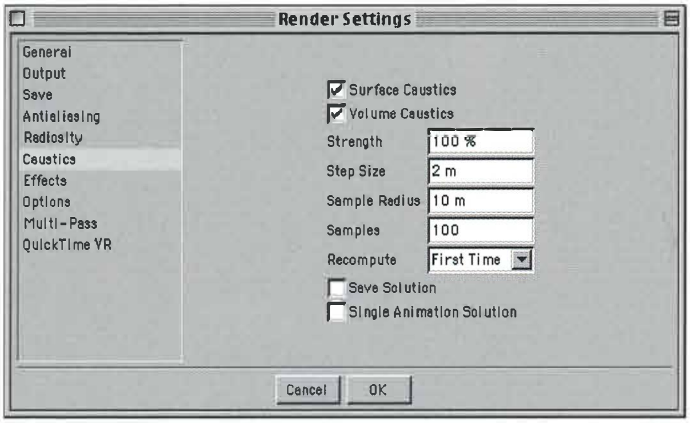
NICKL is an application of these and other functions that can be applied to 3D geometry. NICKL works on primitive, polygonal and spline objects in C4D XL. It can be used to create phenomenal effects. It can be used to modify and animate objects to look like they made from materials that exhibit fluid, elastic and other amazing behaviors. Like all deformers in C4D XL, NICKL is non-destructive. This means that it is applied to a given parent object and can be removed from the hierarchy at any time to return the parent to its original state. This means that you can try all sorts of different settings with NICKL and never worry that your original geometry is being permanently modified or destroyed.

You could use NICKL pretty much anywhere that you want to simulate most any type of fluid effect - like water, fire, gaseous vapor, and various organic movements. It can be used to make great looking water, it can be used to make awesome looking biological animations (amoebae, jellyfish, and more). Applied in small amounts it can create just the right degree of random looking motion that can make an animation come to life.

One of the most powerful features in NICKL is the inclusion of a special class of 3D noise called Periodic Noise. This is a four dimensional tiling noise derived from Perlin Noise. This means that it tiles in the X, Y and Z space as well as over time. Each different axis can be modified to have different periodic cycles and you can tune it to repeat seamlessly in both space and time to whatever setting you require. We'll talk a little more about those settings in a moment.

Getting Started with Caustics

Caustics are turned off by default in C4D XL. There are two places that they must first be enabled. One; any given light in a scene must have either have surface or volume caustics enabled under the caustics property pane, and two; you must also enable surface or volume caustics under the render Settings caustics pane.



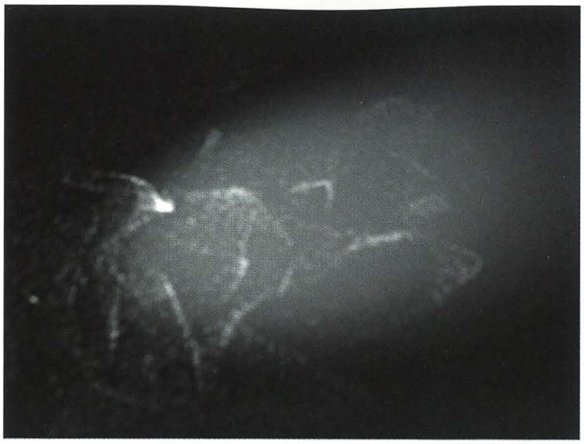
☑ Surface Caustics
☑ Volume Caustics
Strength 100 %
Step Size 2 m
Sample Radius 10 m
Samples 100
Recompute First Time
☐ Save Solution
☐ Single Animation Solution

Having caustics enabled for both a light and the render settings will not necessarily produce any caustic effects. You must first have something for the light in interact with. Additionally the type of material applied to interacting objects matters a great deal too. You will need to make sure that the materials that are to produce the caustic effects have either Transparency or Reflectivity channels activated (See the Tips and Tricks below for issues to watch for).

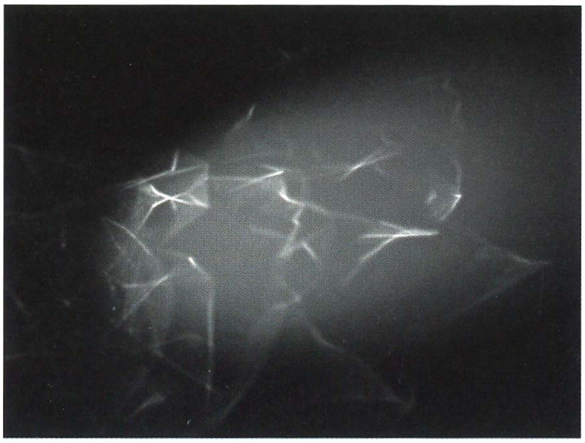
There are three basic settings used with caustics. They are Energy (a percentage value), Photons (an initial starting integer between 10 and 10 million) and Falloff (inner and outer ranges for the effect in scene units).

The default settings for caustics work well when used with a default Spot Light. This is because the photons are being projected down a narrow cone. With the default settings you have 10,000 photons being scattered across a given area and each photon is contributing .01% luminance to the total sample set of photons. This means that each photon is low in intensity and when you get a lot of them being clumped together like you might through a refractive surface, that you will get a decent, though somewhat granular effect. This is because the defaults are configured to give good visual feedback with a minimum render cost.

If you increase the number of initial photons to 500,000 you will see that the resulting effect is much smoother in appearance. This is because with the higher photon count, each photon is now contributing



Default through sample material



Half-million through sample material

.0002% luminance to the total sample set of Photons.

Half a million photons for surface caustics sounds like a lot - this does require around 35MB of RAM at render time for just the Photon map, but it is still reasonable. You can go much higher, but this will take longer to generate and will require more memory. Remember this is for each light which has caustics enabled. It is a good work habit to get what you need from one or two caustic enabled lights.

Also realize that number of photons specified is the initial starting number of photons. If a photon reacts with a reflective or refractive surface an additional photon will be generated, in the same fashion as rays in the raytracing process. Scenes which are highly refractive or reflective and have complex reactive surfaces can see a dramatic increase the final photon count from the initial starting value specified in the light settings. Careful placement and usage will keep render times reasonable and the visual results pleasing.

The previous examples only use a Spot Light. Different light types will require different treatment and handling. See the Tips and Tricks below for details on how each type of light handles caustics.

So far we have only looked at surface caustics. Volume caustics and surface caustics use identical options in the light settings. For volume caustics to work you need to make sure that the Visible Light settings are set to Volumetric. The Visibility pane has settings which control the volumetric effects for the light, and these in turn also control volume caustics. It is also worth noting that the Brightness setting in the Visibility pane also acts as a gain control on the volumetric caustic solution. It is possible that you could have a volume caustic calculated at over 1,000,000%!

The number of volumetric photons needed for a given scene is generally considerably fewer than those needed for surface caustics, but this is dependant on your given scene and requirements. As mentioned earlier, the volumetric photon number represents the initial number of photons generated by a given light.

A falloff setting controls the outer volume depth to be evaluated for volumetric caustics. This is set to 1000 C4D units by default. Different falloff functions are available for controlling the decay of the effect. An inner depth control is also used to offset the initial depth evaluated. With these controls you can confine the volumetric caustics to a given region of your scene, and control the amount of memory required to evaluate and render the effect.

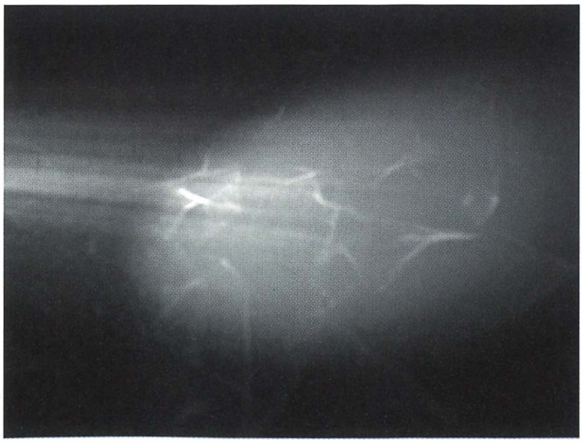
Since this is a volumetric effect, many of the same rules for setting up volume lights apply to volume caustics. The specific controls for how volumetric caustics are rendered are found in the caustics pane in the render settings. As we mentioned earlier there are check boxes for enabling the two different kinds of caustics - Surface and Volume. The Strength setting affects both surface and volume caustics alike. Strength is a gain control that can either increase or decrease the intensity of the caustic effect of the finished render. A setting of 100% is the same as a gain of zero. Adjust this according to your needs.

Unlike volumetric light effects, volumetric caustics do not require that soft shadows be used to generate the effect. In fact volumetric caustics do not require that any shadow be generated at all. This will allow you to utilize this effect in a wide range of settings.

The next three settings; Step Size, Sample Radius and Samples are only editable if volume caustics are enabled. For most situations the default Step Size is too high. You will recall that volumetric effects require a series of samples to be sampled along the Z-axis of the volumetric region. The Step Size controls the spacing between these samples. It is important to note that each step requires its own photon map to be generated. This can mean that very large amounts of memory can be required for a volume caustics solution. Use them carefully. Generally speaking you will want to work with a Step Size which is, at most, set to half the size of the Sample Radius. This will result in a smoother looking application of the effect. This does not mean, however, that you should not try different ratios. By all means experiment! Frequently it is the "happy accident' that leads to innovative discovery!

The Sample Radius will control the local scale of the volumetric caustic effect. A larger Sample Radius will mean a larger volume will be sampled for any given photon. The number of Samples defined will affect the number of photons sampled per step. Lower samples will result in a more "granular" look and higher samples will result in a more homogenous and smooth looking effect.

Materials can also have influence over how caustics are handled. If you click on the Illumination channel in the Material Editor you will



Volumetric caustics

note that there are different attributes for receiving and transmitting caustic effects. These work as a material-based gain control for both the receiving and sending photons. Adjust these as you see fit for your given scene.

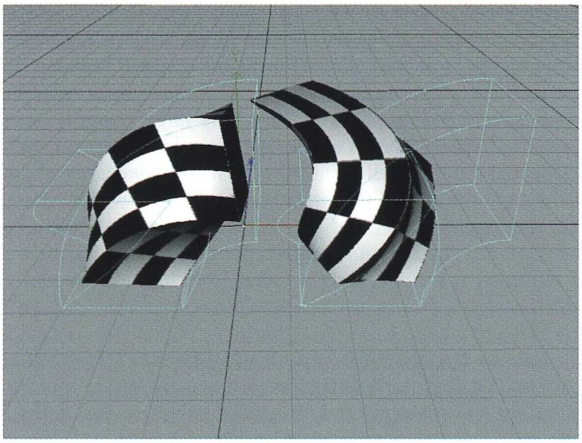
The best thing to do is to get your feet wet and start playing with these features and parameters to see what they do and how they work.

Getting Started with NICKL

Deformation objects or deformers are, in concept, a very simple way to affect objects and groups in your C4D XL scenes quickly and easily. There are a few basic rules to keep in mind so that you can understand and anticipate what sorts of results you might get.

For a deformer to have any effect at all it must be made a child of the object or group that you wish to affect. You can affect many separate objects simultaneously by placing all the objects and the deformer as children of a Null Object or Empty Object. You can even have more than one deformer operating simultaneously on a group of objects. To do this, place everything below a parent object in your hierarchy.

In many cases the order of the deformers in the hierarchy has a direct effect on the resulting deformation. For example having a Bend Deformer above a Taper Deformer in the hierarchy produces a very different result when their hierarchical positions are reversed. It is well worth your efforts to spend some time trying out different arrangements of deformers in the hierarchy to better understand the results.



Deformation hierarchy swap

It is also possible to limit the effect of a given deformer through the use of Vertex Maps, and Selection Set Restriction Tags. Remember that you generate and place the Vertex Maps on the object being deformed and the Restriction Tag on the deformer. You get them to work together by naming Vertex Map and Selection Set in the Restriction Tag. You can also set the percentage of influence a given Selection Set or Vertex Map has in the Restriction Tag.

NICKL is a very flexible and special type of deformer. It is a plug-in and is accessed from the plug-in menu.

When you first create a NICKL object for your scene and double-click it in the Object Menu you will see that it is comprised of five different regions that control different attributes of the deformer plug-in. They are; Function, Attributes, Falloff, Movement and Temporal Animation. Certain function types will disable some of the different input options.

Function:

There are seven different function types to choose from; Noise, Period Noise, Fbm, Turbulence, Wavy Turbulence, Organica (developed by Paul Everett) and None. Some of these noise types should be familiar as they are based on the internal noise functions in C4D XL. Period Noise and Organica are special function types. Organica is a sinusoidal function. Period Noise is a special application of Perlin Noise that seamlessly tiles both in Space (X,Y and Z) and Time. Because of this the entire Temporal Animation section and Period input fields under the Movement section are only active when Period Noise is selected.

Also part of the function controls is a Scale setting for controlling the scale of a given function. As one would expect - a lower scale percentage will result in a small noise and a larger scale will result in a large noise. Scale can be set from 0% to 1,000,000,000% and higher. It should be noted that values over 2 million can result in rounding errors when the NICKL parameter dialog is closed.

Attributes:

Under Attributes there are settings for Direction, what Space the deformer operates in, and the Strength of the deformation. The choices for Direction are Planar, Spherical and Normal. Planar and Spherical are applied as one would expect. Normal, however, is a special case. Normal applies the deformation along the normal of any given vertex. The Space setting determines what space the function is evaluated in.

Object means that the function will behave as if it is always located at the axis of the object. This means that you can have multiple objects being deformed by NICKL and each one will have the deformation applied from its axis - even if they are all in different locations or have different rotation settings. World means that function will behave as if it is always located at the world origin of the scene. Deformer means the function will be evaluated based on NICKL's position and rotation.

Strength settings can be either positive or negative percentages. High settings (over ±1000%) will generally produce extreme results.

Falloff:

Falloff can be set to three different Types (Infinite, Spherical and Cubic). Infinite will result in NICKL being applied to the scene uniformly, Spherical and Cubic will limit NICKL to a region defined by the Radius (in C4D scene units) and will decay from the center of the region to the edge based on the Falloff percentage value. Falloff values can be from 0% to 100%

Movement:

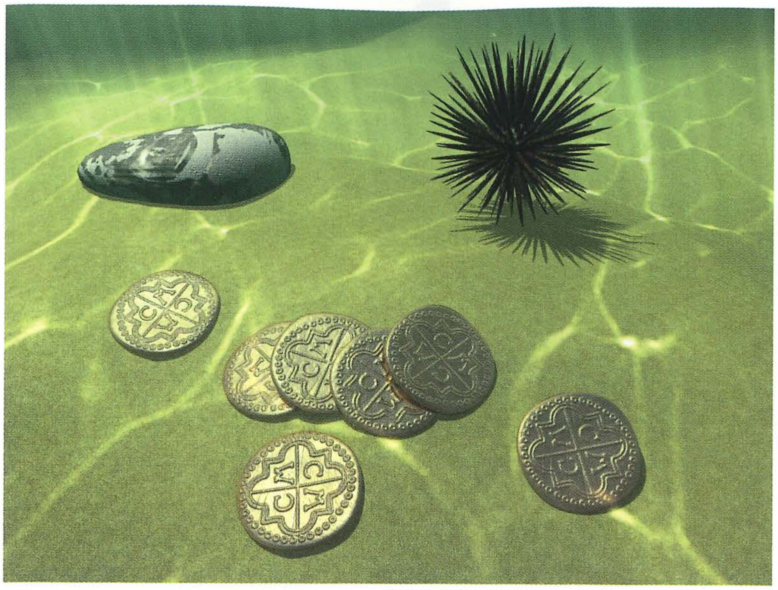
The Movement controls can animate the offset of the function by translating it through any combination of values across the X, Y or Z axis. A value of one represents a movement translation of 100 C4D scene units per second. It is important to note that this assumes that the Function Scale and the Movement Speed are both set to 100%. Adjustments to Function Scale and Movement Speed will act as a gain control on the Translation. For example - if you have your Function Scale set to 200% and you want to have the noise translate 100 units per second, you will either need to set the Translation value to .5 or the Movement Speed to 50%. This can get a little tricky since all these values are all linked together.

Temporal Animation:

This controls the speed that the function is run at over time. The function will animate and deform your objects even if you have not applied any translation within NICKL. Many of the noise functions have very high internal frequencies (Fbm and Wavy Turbulence for example) and will generally look better if you run them at lower Temporal Speed values. The Temporal Animation Period setting is only available with Period Noise. There are some tips listed below on working with periods and getting NICKL to seamlessly loop over time when using Period Noise.

Putting it all together

So now that you know the basics about bhodiNUT's NICKL



Finished image

deformer and C4D XL v7's caustics, the next thing to do is to put them all together in a simple scene that can be used as a seamlessly looping web animation. For this quick scene a handful of gold coins has been lost at the bottom of a tropical lagoon. A sea urchin watches over the sunken treasure and gently waves its spines. NICKL is used in this scene to deform the water surface, creating the surface caustics pattern on the sea floor and to produce the volumetric caustics projecting down into the water. NICKL has also been used to pose and animate the sea urchin. All settings were optimized so that the scene could be rendered as an animation and looped over time as a web viewable graphic.

The best way to see how this was all done is to look at the project file and play with it at your own pace. To download the project files, textures and other goodies referenced in this article please visit: **http://www.lizardlounge.com/keyframe/**

If you do not have a copy of CINEMA 4D XL v7 - you can download a demo version from MAXON's web site along with a copy of NICKL and many other great resources so that you can start working with caustics and NICKL deformations right away.

Tips and Tricks

For caustics to work you need a Transparency with a refractive index setting between .25 and 4.0, but not equal to 1.0. What this means is that C4D XL supports refractive values from .25 on up to a value of 4. A value of 1 applies no refractive change to the rays passing through it and as such will not be used in the generation of any caustic effects.

If you have a reflective material that is 100% transparent and has an index of refraction equal to 1.0 you will need to lower the transparency before the material will be used in the caustics solution at render time. Otherwise objects with this material will be ignored by the caustics portion of the render process.

Even with shadows enabled, the samples for caustics can bleed behind other surfaces. Be aware of this when you construct your models and create your scenes.

Omni and Tube lights project their illumination into a cubic region and there are 6 separate Photon map passes required for each light. Photons are divided between each of these 6 projections. You may wish to increase the Photon count and/or intensity with Omni and Tube lights.

Spot Lights (Round and Square) project their illumination into a conic region. There is only one Photon map pass required for each light. The outer angle settings for the light controls the area of the Photon map.

Parallel Lights, and Distant Lights project their Photons into a cylindrical region defined under the Shadow pane. Use the Parallel Width to control the area of the Photon Map.

Round and Square Parallel Spotlights also project their Photons into a cylindrical or cubic region defined under the Detail pane. Use the Outer radius to control the area of the Photon map with this light type.

Area Lights are a special case. Their size is controlled by the Outer Radius setting in the Details pane, but due to the very nature of Area Lights the region is difficult to predict or define. It is basically the same as the area illuminated by the light and is projected generally along the light's axis.

NICKL

Isoparms and Normal mode will cause NICKL to not be evaluated. This is because there are not any normals to evaluate when working in Isoparm view in the Editor Window.

By default NICKL is configured for Period Noise and is set up to have it loop seamlessly. All the period values are set to "3" which corresponds to a default C4D XL scene having a 3 second duration. For seamless looping the duration of the scene, in seconds, should be divisible by the Movement Translation Period values. Here is a handy table to get you started with this:

Scene Length (seconds)	Periods that will loop seamlessly (Movement and/or Temporal)
1	1
2	1, 2
3	1, 3
4	1, 2, 4
5	1, 5
6	1, 2, 3, 6
7	1, 7
8	1, 2, 4, 8
9	1, 3, 9
10	1, 2, 5, 10 (and so on)

Remember that these values only work with the Scale and Movement and/or Temporal Animation Speed set to 100%. You can achieve other looping durations by adjusting the Scale and Speed settings accordingly. For example; if a scene is 4.5 seconds in length, then set the period to 9 and the speed to 200% and you will loop every 4.5 seconds.

Fun Recipes

Magnet: Use NICKL in what can be referred to as "Magnet Mode". Set Noise to None and Falloff to Spherical. Now move the NICKL region around and watch as it pulls on the object like the Magnet Tool. Remember this is a non-destructive process - so have fun with it!

Flying: Group NICKL and your object to be deformed under a Null Object. Have NICKL with Scale and Strength set fairly high (Scale=1000%, Strength=500%) and set the Attribute for the Space to World. Then set Falloff to None. Now move the object around and watch as NICKL moves it all about. You can make it look like something is swimming or flying through a fluid medium with this technique. This process combined with Vertex Maps can make awesome animated motions!

Continued on Page 42

Selecting a VFX Training Program

by Patricia K. Kelly, Ed.D.

A few VFX professionals are essentially savants — people so gifted that their artistic talents soar on the screen and their work wells up from some unknown internal innate source (think Taron). A few others are fortunate enough to be given oodles of on-the-job mentoring and training, although given the worsening budgetary/deadline stresses within the industry, this situation is increasingly rare. This article is for the rest — all those who want to begin in the field or to upgrade their skills and who must seek out a training program. The options are daunting — there are short-term intensive programs, self-tutorials, on-line choices, opportunities associated with professional conferences, vocational- or trade-type schools, variations on traditional art education, et cetera ad infinitum. How can one compare apples to oranges and make a wise selection?

I am an adult educator, so I offer here some suggestions and guidelines which are rooted in principles of adult learning and adult education. If you are looking at various VFX training programs and trying to select one, or if you are an administrator/teacher in a VFX training program, these ideas may give you food for thought.

Let's begin with what this article is NOT:
* It's not a comparison/contrast of various VFX training programs.
* It's not a recommendation for any specific VFX training program.
* It's not a cost-benefit ratio in terms of finances. (Only you know what your budget allows.)
* It's not a time-analysis of contact hours, clock hours, and lab duration. (Again, only you know what your personal/professional schedule permits in terms of training time.)

An assumption that I am making in writing this article is that all those seeking VFX training are adults. Adulthood is not necessarily defined by chronological age; we all know teenagers with wisdom beyond their years and middle-aged fools who never grew up. Rather, adults are people who take responsibility for their own lives and for those who depend on them. Entering training to begin or to enhance a career generally reflects the act of taking responsibility; ergo, those engaged in this activity can be viewed as adults, regardless of age. It is particularly important for those involved in planning and providing training to bear in mind that the learners with whom you're dealing are, indeed, adults. In our culture, with its emphasis on public education, we tend to automatically think of learners as children; so working with adults may require changing your thinking about who those eager faces are in your classroom. As you look at your own training program (as a facilitator) or one you're contemplating joining as a learner, ask these questions:
* Are all learners viewed as adults?
* Are learners treated as adults, with "real lives" outside of the classroom? (e.g., Does the trainer acknowledge and reasonably accommodate for "life stuff" such as sick spouses, child care issues, car breakdowns, unanticipated overtime, etc., etc.?)
* Do trainers treat one another with collegial respect and extend that respect to the learners, essentially as "junior colleagues" in terms of experience within the field but as equals in terms of life-experience as adults?
* Do trainers use language that indicates their understanding of learners as adults (e.g., referring to a classroom full of people as "learners" or "folks" rather than as "kids")?
* Are the learners presented with the kinds of challenges and reality exercises that will be required of them in the near future as adult job-holders in the area of expertise being taught?

Look for programs where the answer to all of these questions is an unqualified, "yes!"

Andragogy has been defined as "the art and science of helping adults learn" (Knowles, 1980, p. 43). Many of us are more familiar with the term pedagogy, which refers to teaching children. Andragogy is the parallel for working with adult learners. When Malcolm Knowles developed the definition of andragogy within the field of adult education, he noted that there are five basic assumptions or characteristics of adult learners. For those who want to learn more about this theory, check out the references below. For now, let's look at these five characteristics of adult learners and how each of them might apply in a VFX training program.

1. As a person matures, his or her self-concept moves from that of a dependent personality toward one of a self-directing human being. Children allow others to make decisions for them — what clothes to wear, what food to eat, what time to go to bed, etc. Adults make their own decisions. Keeping this in mind, here are some questions to consider when looking at a VFX training program:
* Are the learners given choices within their programs (what classes to take, what projects to do, the subjects of their demonstration work, etc.)?
* Are the learners given clear explanations of the choices available to them and the consequences of each?
* Are the learners' decisions respected?
* Does the staff listen to suggestions made by the learners concerning the shape they want their programs to take?
* Can training be individualized to fit the needs of each learner?

Again, seek VFX training programs for which the answer to all these questions is, "yes!"

2. An adult accumulates a growing reservoir of experience, which is a rich resource for learning. Let's pretend that a client wants you to model and to animate a dog for a pet treats commercial. Think about everything that you know about dogs — all the different breeds, how they look, how they move, what shapes and sizes and colors they are, what behaviors they exhibit, etc., etc. You will bring all that information to bear in

doing your work. Ask a young child about dogs, and you will quickly see that you, as an adult, have far greater depth and breadth of knowledge, even on this simple subject, to draw upon. Developers of good adult education programs and exemplary trainers realize that their learners have a tremendous wealth of experience and knowledge to bring to the table. Check out potential VFX training programs by asking questions like these:
* Are learners encouraged to share their backgrounds (educational and/or work experience) in the classroom when appropriate?
* Are learners with particular areas of expertise ever tapped by the trainers to share additional information with the group/class as a whole?
* Do the trainers respect the learners' "former lives," that is, the work that the learners did prior to the current training?
* Do the trainers ask the learners questions about their existing knowledge on a topic before plunging into new material?

"Yes" answers to these questions indicate that the VFX training program understands this aspect of adult learners.

3. The readiness of an adult to learn is closely related to the developmental tasks of his or her social role. If you are currently employed in the field of VFX, or if you hope to be so employed soon, your readiness to learn the material covered by a VFX training program is likely very high. If you are hoping to improve your current working conditions and/or to raise your salary so you can better support yourself and your family, your readiness is probably even higher. VFX training programs which recognize this readiness are worth exploring. Here are some questions to ask in finding out if the program you're considering has this quality:
* Have you been asked by the program facilitators what you plan to do with what you'll learn in the program?
* Can the program be modified to fit different needs and plans for different learners?
* Does the program accept only those who show talent and/or demonstrate willingness to work hard (as opposed to every applicant who can pay)?
* Do the trainers know what other roles you play in your life (spouse, parent, civic/religious organization leader, sibling, friend, etc., etc.) and how those roles interact with your plans to use what you're learning in the VFX training program?

Once again, look for "yes!" answers to these questions.

4. There is a change in time perspective as people mature — from future application of knowledge to immediacy of application. Thus, an adult is more problem-centered than subject-centered in learning. Ask most junior-high-school students what they're studying in school, and they'll give answers like, "math, science, and social studies." Kids tend to think of learning in terms of subjects — 1st period is Music, 2nd period is English, 3rd period is History, and so forth. On the other hand, adult learners focus more on what they'll accomplish through the learning. For instance, one adult learner may state that his goal upon completing a VFX training program is to move up the ladder from in-betweener to supervisor; another may share that her goal is to gain the skills needed to leave a large production house and begin working as an independent contractor. Those kinds of responses to the inquiry, "What do you hope to get out of your VFX training program?" are much more common than, "I want to be able to use software package XYZ." And, even if the latter is the response, a little probing will usually show that using that software package is the means to a bigger end (e.g., by mastering that software package, the learner can now finish a personal project that's been hanging fire for awhile). VFX training programs which can look beyond subject-based training and help learners plan for their bigger goals and dreams are worth finding. Here are some questions to help you in your search:
* What types of VFX-industry jobs have people had before and after completing this program?
* What will completion of this program enable me to do that I can't do now?
* Let me tell you my goal: It's ________; how will this program help me get there?
* What kinds of projects and demonstration materials will I create during this program?

Obviously, in this set of questions, you're not looking for yes/no answers but for information that will help you determine if this VFX training program matches your inclinations and if it will fill your needs. Although you should expect every program to clearly outline the subjects and topics covered during instruction, steer away from "canned" programs that go no further. Seek out programs that have a broader scope and a sense of a "bigger picture" than just completing a set of click-and-drag steps.

5. Adults are motivated to learn by internal factors rather than external ones. We all respond to external factors sometimes. The praise received when winning a prize for our work (not to mention the cash award!) feels great. However, adults are rarely grade-driven in classes (What does an A or a B in VFX mean anyway?); the sense of having completed a job well is much more important to most of us than are arbitrary letters attached to our accomplishments. VFX training programs must have standards and all learners must attain certain specified competencies to demonstrate mastery of the material; in some situations, letter or numerical grades (e.g., percentages on written tests) may be used in monitoring these standards and competencies. It's a necessary part of evaluating both the learners and the program. However, the overall atmosphere of a truly adult-oriented program will minimize those external, "objective" measures and will optimize the encouragement of each learner's potential. Here are some questions to ask in looking for that atmosphere in a VFX training program:
* What kind of feedback is given to learners?
* How are critiques handled?
* If I'm not performing up to expectations, how will I be told?

Continued on Page 42

STUDIO QUALITY LIGHTWAVE TRAINING GOES ONLINE

On August 20, 2001, online LightWave 3D training courses will be offered through three renowned schools currently teaching CGI animation: Foundation Institute (a Division of Foundation Imaging), 3D Exchange, and Silicon Valley College. This three-school collaboration was the end result of an ever-increasing demand for professional instruction in 3D Computer animation.

According to Karen Tosoni, owner and founder of 3D Exchange, and currently Program Director at Foundation Institute, “In order to meet this demand we have decided to go worldwide with instructional training, and offer our proven and specialized LightWave courseware online.” Tosoni also mentioned that “it's a wonderful time to assist individuals seeking professional training, and instructors who need both structured courseware and resources for their students.”

Not only will students learn the LightWave 3D software program, but networking skills, industry leads; software upgrades, and changes within a particular industry as well. In addition, the on-line program will also offer a demo reel evaluation and job placement. assistance.

What qualifications are required of an individual wishing to take online training? Tosoni explains, “We don't require a portfolio, or a demo reel. We are more interested in someone who demonstrates a desire to excel, is self-motivated, dedicated, and able to set realistic goals. LightWave is powerful, complex, and wonderfully versatile. It simply not a program you can really learn, much less master, in a few months. We are providing tips and tricks from professionals to excel the process, but it is really up to the student to decide if this program is for them. And, it is the identical training and courseware you would receive if you attended any one of these training centers in person. You are also afforded the same access to all instructors.”

These three schools were in agreement that the on-line training should be void of promoting false expectations or “sugar coating” job prospects. Rather, they want students to confront the reality of what happens when a job is completed, and learn the various methods to research their industry. How to “understand” the industry, and learn the difference between, 3D -2D, compositing, editing, LightWave and Maya, gaming versus the corporate and medical industries. Don't waste your money.Once you make your decision then it is easier to reach your goal, and a job!

But Tosoni is also quick to point out that on-line training isn't for everybody, “if you're an individual who needs personable interaction then this type of training is probably ill advised, and you should consider the individualized one-on-one, on-site training we also offer. But, you might wish to test the program prior to registration. Take some basic courses online such as industry research, basic LightWave, a workshop, or demo reel evaluation prior to making a full-time commitment.”

The first on-line class will begin August 20th. For additional information: www.foundationinstitute.com or www.3dexchange.com

Get Connected

with Online LightWave Training

Channels
Expressions
Camera
LGLogo_xpar
Light
Adaptive Sampling
Affect OpenGL
Lens Flare
Volumetric Lighting
Affect Specular
Affect Caustics
Lens Flare Options
Light Type
Spotlight
Spotlight Cone Angle
Spotlight Soft Edge Angle
Zoom In
Zoom Out
Reflectivity
Fresnel-like Angular Variation
Minimum Glancing Angle
Rename
Shaders
Remove
Stereoscopic
Image Area
Frame Size
Pixel Aspect
Limited Region
Adjust Region
Camera Mask

We at **3D Exchange** and **Foundation Institute** are proud to announce the creation of our ***Online Curriculum.***

Now we're as close as your own keyboard, no matter where in the world you are.

www.3dexchange.com
www.foundation-i.com

Foundation Institute

3D Exchange | Animation Training

NewTek
Authorized
Training Centers

eality Check (continued)

*Will there be opportunities for remediation in any areas in which I flounder?
* Can my program be accelerated or slowed if I'm achieving well in a certain area or struggling in another?
* Is there more emphasis on learning to do what I do well even better, or on bringing up to speed those areas in which I lag?
* How is "success" defined in this program?

As before, the answers won't be simple yes/no responses, but the information gathered with these questions will give the inquirer a strong sense of the atmosphere of the training program.

A host of factors go into selecting a training program. Certainly, economics, location, and time commitment weigh heavily in the decision-making process. With all the options out there, however, it is worth doing some detailed research to find the right VFX training program for you. Be sure to include at least some of the questions above in your investigation, and it's more likely that you'll discover a training program that treats you like the adult you are.

References:

Knowles, M.S. *The Modern Practice of Adult Education: From Pedagogy to Andragogy.* (2nd ed.) New York: Cambridge Books, 1980.

Knowles, M.S. *The Adult Learner: A Neglected Species.* (3rd ed.) Houston: Gulf, 1984.

Merriam, S.B., & Caffarella, R.S. *Learning in Adulthood: A Comprehensive Guide.* San Francisco: Jossey-Bass, 1991.

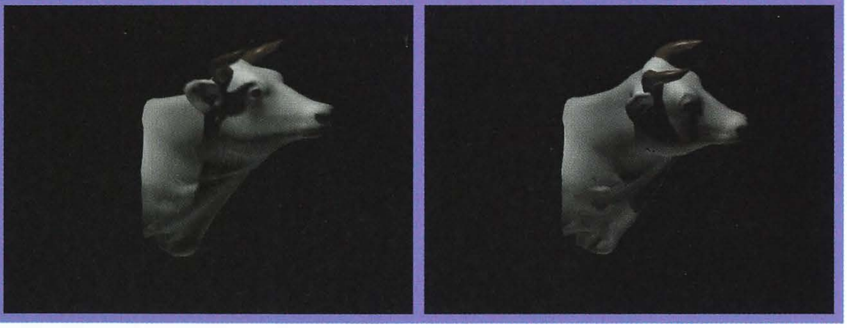
Patricia K. Kelly, Ed.D., is a consultant in adult education & blind rehabilitation living with her animator husband whom she adores in Maine. Dr. Kelly is available to consult with VFX training programs on their adult education practices, program evaluation, curriculum development, and similar topics. She can be reached at dakelly@megalink.net.

Image sequence from
Motion Designer Nuts and Bolts

Tutorial begins on Page 44

Secondary Motion for Character Animation

by Michael Reiser



Continued from Page 37

Endnotes

[1] As referenced on the University of St. Andrews, Scotland web site: **http://www-history.mcs.st-andrews.ac.uk/history/Curves/Definitions2.html**

For more historical information on Caustics, Mathematics and related Optical information please visit an excellent reference on Hamilton's papers on Geometrical Optics as published by D.R. Wilkins' of the School of Mathematics at Trinity College, Dublin web site:
http://www.maths.tcd.ie/pub/HistMath/People/Hamilton/Optics.html

Additional contemporary information as it pertains to Computer Graphics can be found in a paper written by Henrik Wann Jensen, titled "Rendering Caustics on Non-Lambertian Surfaces". A copy can be found on the Internet at:
http://graphics.lcs.mit.edu/~henrik/papers/gi96.html

[2] For more information on Ken Perlin and Perlin Noise you can visit Ken Perlin's home page at: **http://mrl.nyu.edu/~perlin/**

There is a lecture which Ken Perlin gave at the GDC Gamers Hardcore workshop in 1999: **http://www.noisemachine.com/**

Additionally there is a great visual guide to Perlin Noise on Hugo Elias' web pages found at:
http://freespace.virgin.net/hugo.elias/models/m_perlin.htm

*Christopher Mills is one of the founders of Lizard Lounge Graphics, **www.lizardlounge.com**. You can email him at **cmills@lizardlounge.com**.*

Motion Designer: The Answer to Secondary Motion in LightWave 6.5b Character Animation

by Michael Reiser

One thing that can tip you off to a pretty good animation is the animator's addition of secondary motion. Secondary motion is the movement of such things like bouncing alien antennae or the flopping of big ears as a creature turns his head. These motions can be difficult to believably recreate and so are left out of most beginners' animations. LW has the answer and that answer is Motion Designer.

Motion Designer is a nice addition to an already powerful LW arsenal. It is very easy to use and gives great results. Its main use appears to be geared mainly toward simple cloth simulation and Jell-O cubes. It is much more powerful and versatile then that however. Motion Designer is the jiggling jelly belly; flowing tentacles and double chin dream come true! These once difficult secondary motions are now easy. In this tutorial I will be going over how Motion Designer can help add that extra bit of realism by the addition of secondary motion to your characters.

The manuals do a fairly good job of describing what each of the Motion Designer buttons do, so I will not be going into great detail about all of the individual options. Please refer to the manuals for further explanation if needed.

The point of having secondary motion computed for you is to free up your creative talents and allow you to concentrate on your character animation. Some people will say that it is a cheat. And that the animator should be animating the bouncing belly as well as the walk. Yes in certain situations when the secondary motion of a big fat bouncing belly is part of the character personality maybe the animator should have a part in its creation. But should the animator be tied down and bothered with the subtle sway of a creature's ear as it turns its head? When your boss is breathing down your neck wouldn't you much rather have LightWave do all of that for you? Motion Designer is the answer.

Here we will use the good old LW Cow object to show how MD can add a large amount of realism to your animations with just a few simple additions to your normal workflow. In this particular example the secondary motion is the swaying of the redundant skin under the cow's neck and chin as she turns her head. (Yes, it's a "she".)

To start, we need to visit Modeler. MD works with surfaces. In modeler we will define the surfaces for which want secondary motion applied.

In my example (**Figure 1**) you will see that I cut off the head of the sacred LightWave cow. (It's OK, she's not a real cow. This simply kept the render times down and was easier to work with.) The area that makes up that fleshy portion of skin that hangs down below her neck is that area that we will eventually add secondary motion too. I also took the liberty of remodeling this area to make it a little more pronounced to make the effect slightly more obvious. We need to select the polygons that make up this area. Try to imagine where you want the motion to be. Don't choose to large of an area or will look like the Cow is swimming in its skin. Try and choose just the area that makes up the loose fleshy portion of the neck, **Figure 2**.

Note: Here I did not include the polygons that would have connected to the rest of the cow. The reason I did not include these is because otherwise these have no anchor. They would simply swing freely and I simply did not like the way it looked.

These polygons already have a Surface applied to them. We need to declare another surface for MD act on. To create a new surface select the Detail tab and choose Surface from the options list. Here, rename the selected Surface to something like Flabby. Don't change any of the other Surface attributes. You want this surface to blend with the rest of the cow texture. Exit out of the Surface menu.

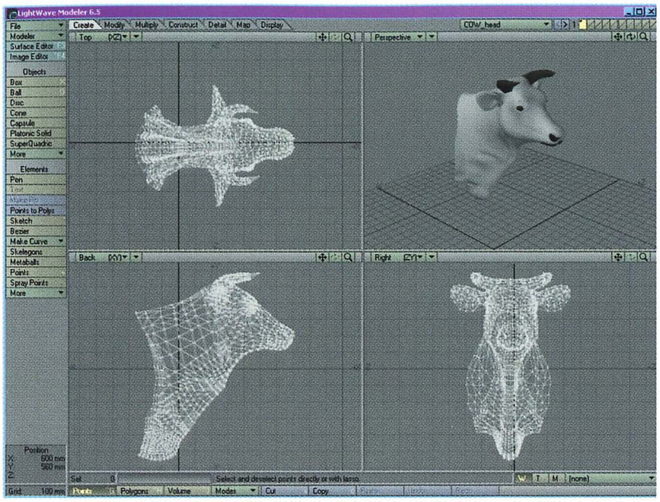


***Figure 1:** Cow head with redundant neck skin.*

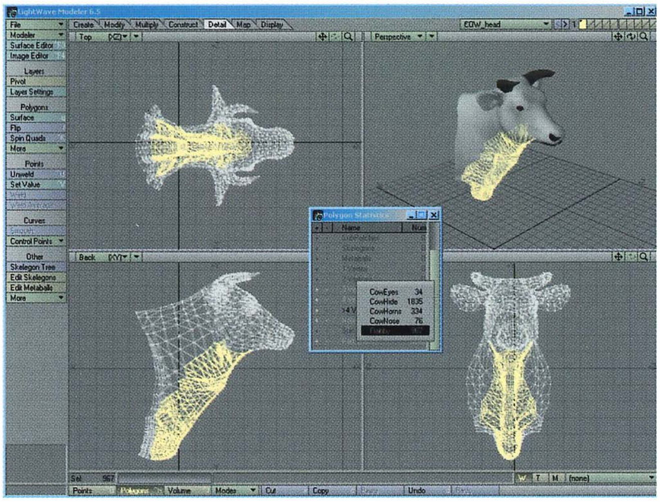


***Figure 2:** Selected polygons.*

Open the Surface Editor and copy the Cowhide texture to the new Flabby texture by using the copy and paste option, **Figure 3**. This way the two surfaces will be seamless.

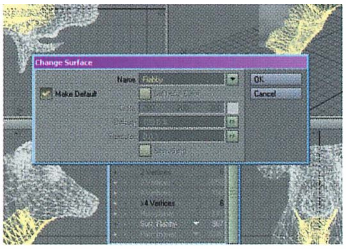


***Figure 3:** Surface selection.*

Depending on the type of motion you intend to have, the polygon count may need to be increased and tripled. This area will eventually be bending and undulating. If the polygons are too few and are not tripled, you may obtain undesirable results and rendering errors. This said, there is an option that allows the use of four point polygons in Motion Designer. I will discuss that a bit later. For now, subdivide and triple the selected polygons. To do this, go to the constructs tab and choose the Subdivide and Triple commands, **Figure 4**.

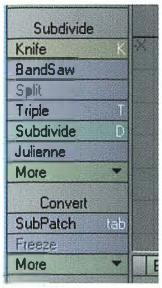


***Figure 4:** Triple and Subdivide under the Construct tab*

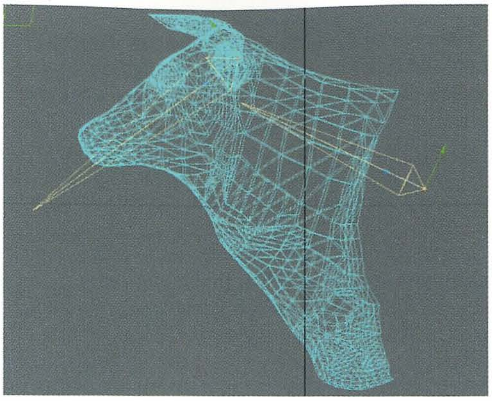


***Figure 5:** Cow's head with bones added.*

Those are the only changes to the model you will need to make. Had you modeled the cow yourself there would have only been the consideration of what surfaces to create and their appropriate subdivision. Not too much extra work considering the payoff. Save the object as something like Cow_Head.

Load the object into Layout. The first step is to add a few bones, one in the neck and its child within the cow's head. Select the Cow head object and open the Add menu from the Item Tab. Select the Add bone option. Move and rotate the bone so that it sits close to where you would expect the cervical spine of the cow to be. Key it at frame zero. Now, form the same Add menu; select the "Add child" option. Move and rotate this bone so that it sits as shown in Figure 5. Key this bone at zero as well. Leave the rest of the settings as is and activate each by hitting the R key on the keyboard, or activate them through the Bones panel.

Note: This was a quick and dirty bone setup. In this tutorial we will not be going into depth on the addition of bones and how they work. If you need any help with bones please see the manual.

Select the Head bone and rotate it, **Figure 6**. The head should move form side to side with some influence on the motion of the redundant skin in the neck. If needed you may have to decrease the strength of the neck bone a bit. Ideally, you want the rotation of the Head bone to sway the skin in the neck from side to side as the head turns. Motion Designer will use the movement of the neck points and polygons as reference to calculate the secondary motion. If the points don't move then there will be very little secondary motion.

Adjust the strength of the neck bone in its properties panel; decrease the spinner from 100% to 50%, **Figure 7**. You may have to play with this a bit. This particular setting worked well for me.

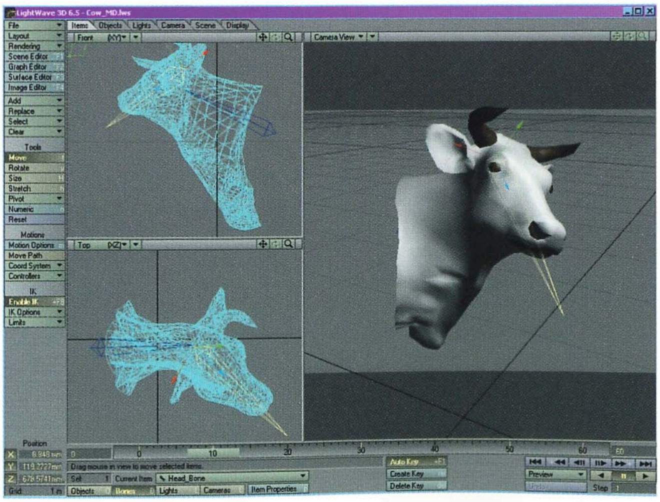


***Figure 6:** Keyframing the cow head.*

Make a number of rotation keyframes so that the cow looks from right to left a few times. If you have not already done so, Save the scene.

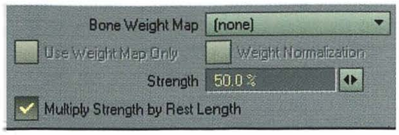

Bone Weight Map (none)
☐ Use Weight Map Only ☐ Weight Normalization
Strength 50.0 %
☑ Multiply Strength by Rest Length

***Figure 7:** Neck bone strength adjustment.*

Open the Scene tab and select the MD plug-in form the selection list. Don't let the size of the pop-up menu, **Figure 8a**, deceive you, this is one of the most powerful plug-ins in Light Wave.

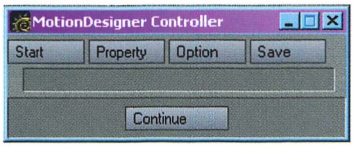


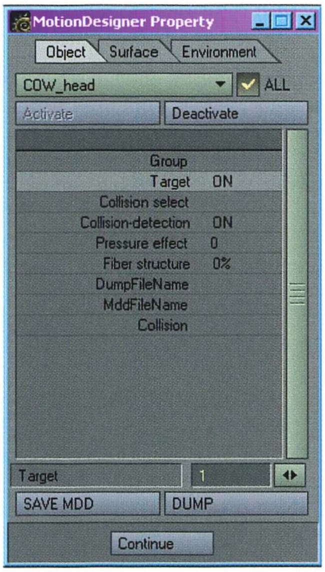
***Figure 8a:** Motion Designer Controller.*

Note: As I stated before the manuals do a pretty good job of describing most of the MD button functions. I will take you through the available options that work well for reproducing character type secondary motion. The rest of the buttons and spinners I leave up to your free time and exploration.

Click the Property button.

Motion Designer needs to know what objects will be included in its simulation. From the drop down list select the Cow_Head object, **Figure 8**. Mouse click once next to the Target option in the drop down list. In a second or two, "ON" will appear. You have just told MD that the Cow surfaces will be taking part in the simulation.

In the real world we have gravity and wind that help with secondary motion and resistance. You need to tell LightWave how much wind and gravity you want. Click on the Environment Tab in MD. Here you will see the options for adding environmental conditions such as wind and gravity.



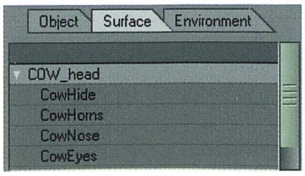
COW_head ☑ ALL

Group
Target ON
Collision select
Collision-detection ON
Pressure effect 0
Fiber structure 0%
DumpFileName
MddFileName
Collision
Target 1

***Figure 8:** Activating the Cow_Head object.*

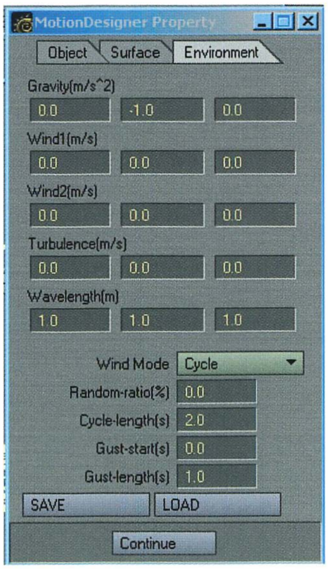
For our purposes the only thing we need to add here is a little gravity. The boxes are not labeled however, they correspond to the X, Y, and Z coordinates of world space. Change the gravity Y direction to negative one as shown in **Figure 9**.

Leave the remainder of the options as is and select the Surface tab. Here you see the names of the activated objects, in this case, Cow_Head, **Figure 10**. All of the surfaces of the Cow are listed (Flabby is too, its just at the bottom of the



COW_head
CowHide
CowHorns
CowNose
CowEyes

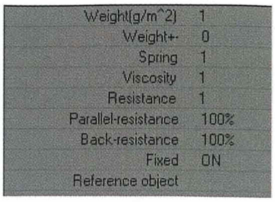
***Figure 10:** MD Surfaces.*



Gravity(m/s^2) 0.0 -1.0 0.0
Wind1(m/s) 0.0 0.0 0.0
Wind2(m/s) 0.0 0.0 0.0
Turbulence(m/s) 0.0 0.0 0.0
Wavelength(m) 1.0 1.0 1.0
Wind Mode Cycle
Random-ratio(%) 0.0
Cycle-length(s) 2.0
Gust-start(s) 0.0
Gust-length(s) 1.0

***Figure 9:** Environment options.*

list.) The only surface we wish to have move is Flabby. The rest of the surfaces we want to stay "fixed" in position. To tell MD this, click on the CowHide texture. A set of parameters will appear. Ignore most of them for right now. Scroll down the list until you see the Fixed option, **Figure 11**.



Weight(g/m^2)	1
Weight+-	0
Spring	1
Viscosity	1
Resistance	1
Parallel-resistance	100%
Back-resistance	100%
Fixed	ON
Reference object	

Figure 11: *Fixed.*

Set the fixed option to ON for all of the cow surfaces except for Flabby. When you are done. Select the Flabby surface and bring up its option list.

Now you may be looking at all of the options thinking that there are too many choices and this is best left to the special effects guys. Don't worry you don't have to play with all of the settings. You may not have to play with any. There are presets, **Figure 12**, that may work for a particular project.

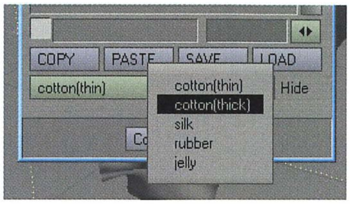


Figure 12: *Preset options.*

Try each of these and see how the cow skin reacts. Look and see what parameters change when going from material to material. Notice how the weight, compress stress and friction are the only parameters that change between thin and thick cotton. Now take a look at the difference in silk. Here, in addition to the same parameters for cotton, spring, stretch limit and sub structure help define silks properties.

To apply the MD settings and see the animation, simply click "continue" then "start" from the main MD pop-up menu.

Note: If you can see that your settings are not what you wanted hit the Control key to stop the calculation.

I think of the redundant skin on our cow as something like leather or perhaps a thick burlap sack. The closest preset to this is thick cotton. Select the Thick Cotton preset and adjust some of its defining parameters to make it more thick and tough. As we mentioned before the differences between thin and thick cotton where the weight, compress stress and friction. Let's play with these and see what happens.

Some changes can be very subtle. To demonstrate the changes of each adjusted parameter, make the changes a little larger then you normally would. You can tone it down after you are sure of the parameters function.

The first parameter is the weight option. Leaving all of the other settings the same, change the weight to 8000 and see what happens.

Figure 13 shows the drooping of the redundant skin under the effect of increased weight. It now has a taffy look to it and reacted slower to the head motion. For all intents and purposes this parameter does what it says. Return the weight to something like 20.

The next parameter that differs between thin and thick cotton is the Compress stress. Let's see what happens when this is decreased.

The effect of decreasing the stress to zero is more of a dynamic look and feel. (It's not easy to show this change in a single image, **Figure 14**.) But with zero Compress stress the skin waves like a thin piece of cloth. Increase it and it holds its more of its original structure. For the cow I choose to keep the Compress Stress at 100%. This will give the feel of a thicker more stiff material.

The next of the differing parameters is Friction. This really has a large influence when a material is interacting with another material. Much like the good old cloth draping over a ball routine. In our situation there is not much of a contribution from the friction setting. Friction would perhaps have some influence on the skin folds if the surface is double sided and self-collision is active. I however did not get good enough results that warranted the extra computational time.

I have found that the two factors with the largest influence on the apparent stiffness (In addition to the compress stress setting mentioned above) are the Spring and Sub structure parameters. In my final render I set each of these to 2000 to get my desired results. Maybe over kill, but it worked. Try going to the other extreme and set them both to zero. In this situation there is nothing to hold the surface to the cow and the flabby polygons drop off due to the gravity settings.

When it comes down to it, the presets, even though there are only five, do a pretty good job of getting you started. Use them as an initial guide to the type of material you are looking for. Take one parameter at a time and adjust it accordingly. In the case of character animation and secondary motion, most of the time you will be looking to add a stiff flesh like effect. (The Cows dewlap in our example.) For flesh I suggest building from the stiff cotton material and increasing the Spring and Sub structure until you get the feel you are going for.

I experimented quite a bit with Motion Designer over the past few months. I went through multiple models and attempted numerous types of materials. The hardest thing yet is the reproduction of clothing. Polygon shirts, and the like, seam to jump and slide, giving less then desirable results. Cloths are not a complete loss however. I found that if clothing is modeled in the more tradition sense MD could be applied to shirt cuffs and coattails in much the same way we set up the cow example. The uses for Motion Designer really go on and on, squid and jellyfish tentacles can use some help from MD as well.

As I wrote this, the manual (hi-res version) was open the entire time. There are good explanations for most of the parameters and I suggest you use it. My intention was not to rewrite the manual but to suggest another use for an already great plug-in. I hope that I have sparked a few ideas and gotten some creative juices flowing. Its time to move past flags and square pieces of cloth.

Michael Reiser is one half of a two-man creature and character creation team. He and the other half, Doug Dworkin, collectively make up Pollywog Productions, a small animation house located in Pittsburgh, PA. Any questions you may have about Motion Designer or just want to discuss the how to get that Sculpey smell out of the oven, feel free to write. ***MikeReiser@aol.com***

Image sequence on Page 42

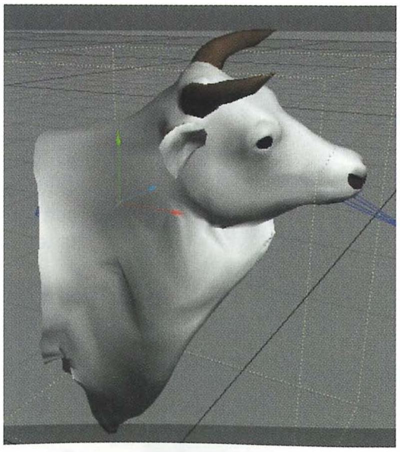


Figure 13: *Weight increased.*

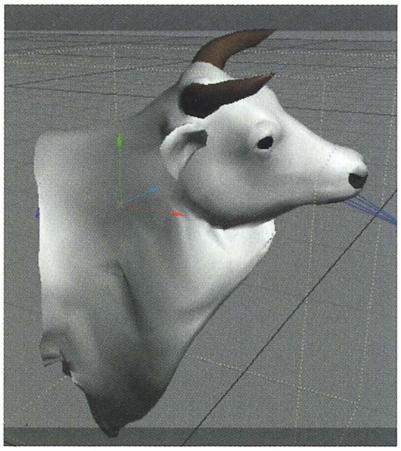


Figure 14: *Decreased Compress stress.*

Architectural Photography

by Jared Bendis

It is hard to shoot good architectural photography. People never realize just how hard until they try. Buildings are big. I love to take pictures of buildings, especially castles (which are very big!). Take a clean, undistorted, and perfectly aligned shot of a building, map it onto a box and you've just erected a building on your virtual set.

Normally, if you want to take a picture of something big, you just back up. The problem is, sometimes there isn't enough room to back up. Worse, when there is enough room to get the whole building in view, you find that everything is in front of it. Trees, signs, people, and other buildings will obstruct your view and prevent you from getting a clean shot, **Figure A**.

You also have to make sure that your camera is level and parallel to the plane of the building. A tripod (if they let you have one onsite) can keep you level with the building but keeping yourself parallel with the plane of the building (both left/right and up/down) you have to do on your own. Sometimes you have to purposely tilt the camera up just to get the whole building into view.

This tutorial outlines how to capture and correct a clean and aligned shot of a building. As always there is more than one way to skin a cat, this is just the technique that I have found that works most effectively in terms of both time and money.

Hardware: The Wide-Angle Lens

The human eye is roughly equivalent to a lens with a focal length of 50mm (which is why most 35mm cameras are equipped with one). The horizontal field of view or FOV of a 50mm lens is about 40° (the vertical FOV is about 27° because 35mm is a rectangular format). Lenses with a focal length of about 40mm and higher are generally referred to as rectilinear lenses. Rectilinear lenses capture images in the same way a human eye sees, and lines that appear straight to the eye also appear straight with a rectilinear lens.

To get a wider field of view you need to use a lens with a smaller focal length. Wide-angle lenses can go down as low as 14mm (which offers a horizontal field of view of about 105°). Typical wide-angle lenses for 35mm cameras are 35mm, 28mm, and 18mm. Wide-angle lenses capture more than the human eye but the lenses also produce linear distortions. These linear distortions cause straight lines to appear curved depending on the focal length of the lens and the location of the line within the image.

A fisheye lens (typically with a focal length of 8mm) is an extreme wide-angle lens that causes all straight lines to appear curved. A fisheye lens produces a circular spot on the rectangular image, **Figure B**. The field of view of this spot is generally 180° or more.

Another type of lens is called a full frame fisheye. A full frame fisheye fills the entire rectangle (no black space like on the spot) and has a diagonal Field of View of 180°, **Figure C**. A full frame fisheye is like zooming in and cutting out the largest rectangle from a spot fisheye.

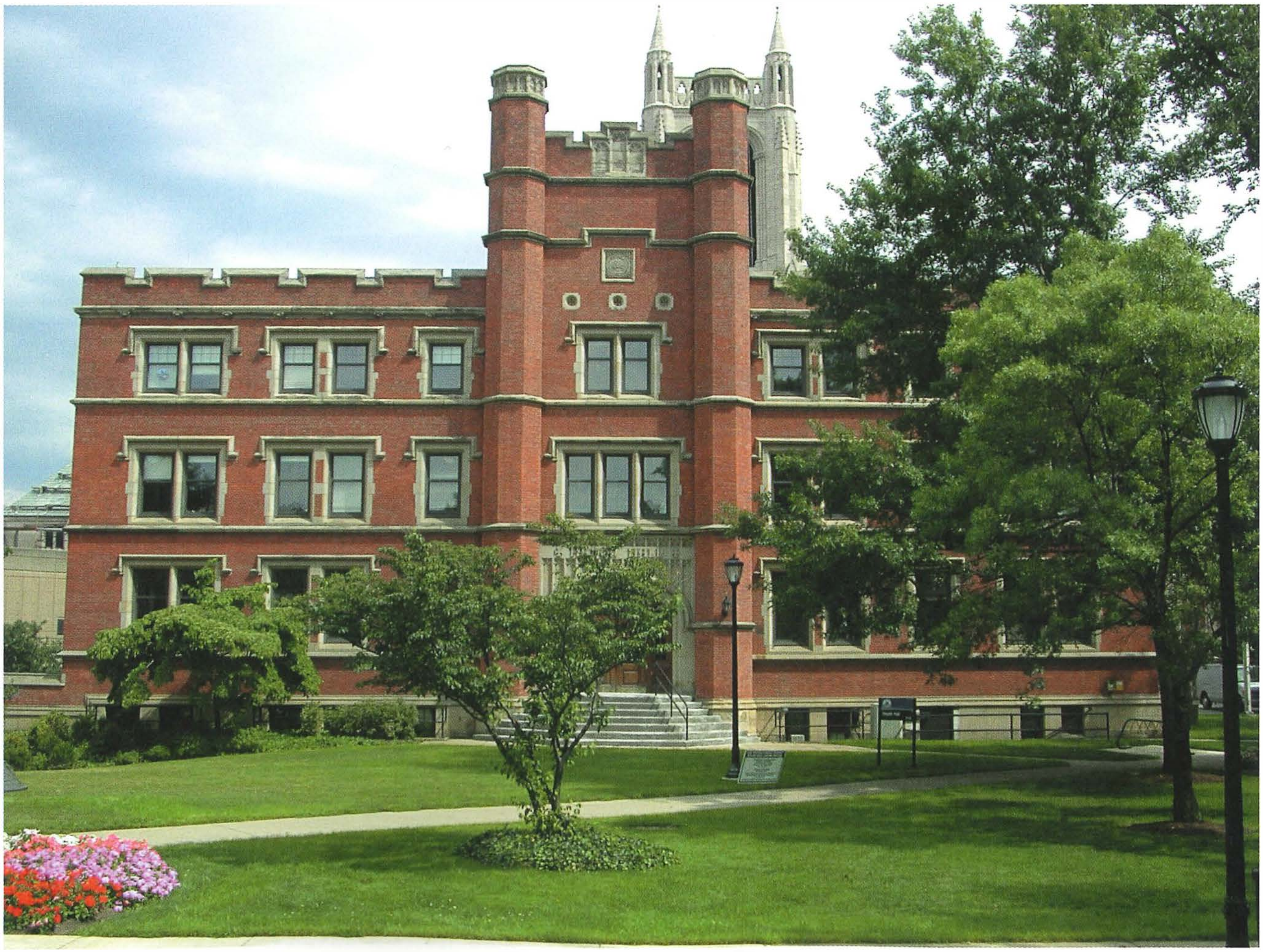
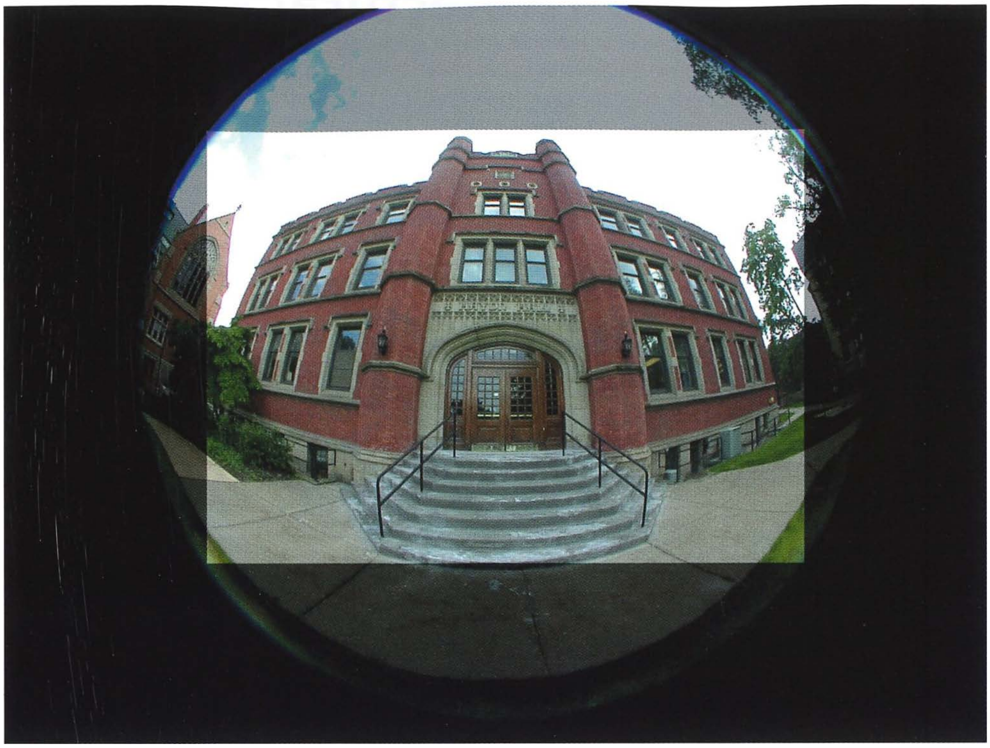


***Figure A** - 38mm - 50° FOV*

To shoot architectural photography you need to pick a wide-angle lens that fits your budget. I use a Nikon 880 digital camera with the 8mm fisheye converter lens (and the required step-down adapter ring). I could also use my 35mm Pentax K-1000 with my 18-28mm wide-angle zoom. However I prefer to use my digital camera when doing this much post-processing to take out the additional steps of developing and scanning and also because I can get a wider field of view with the 8mm fisheye converter lens.

The Nikon 880's built in rectilinear lens zooms from 38-115mm and in addition to the 8mm fisheye converter lens there is a 24mm wide-angle converter lens (as well as a variety of telephoto lenses). The Nikon 880 is a 3.34 megapixel camera that uses Compact Flash memory cards. The camera is great for film photographers. It has aperture and speed controls (though only 2 aperture settings) as well as settings for white balance, ISO rating, and others. More importantly it allows you to save images as lossless TIFFs (though they are 9MB each). The camera body is very small and easily fits in the palm of your hand (the adapter and fisheye lens are practically the same size as the camera itself).



Full Frame within Spot Fisheye

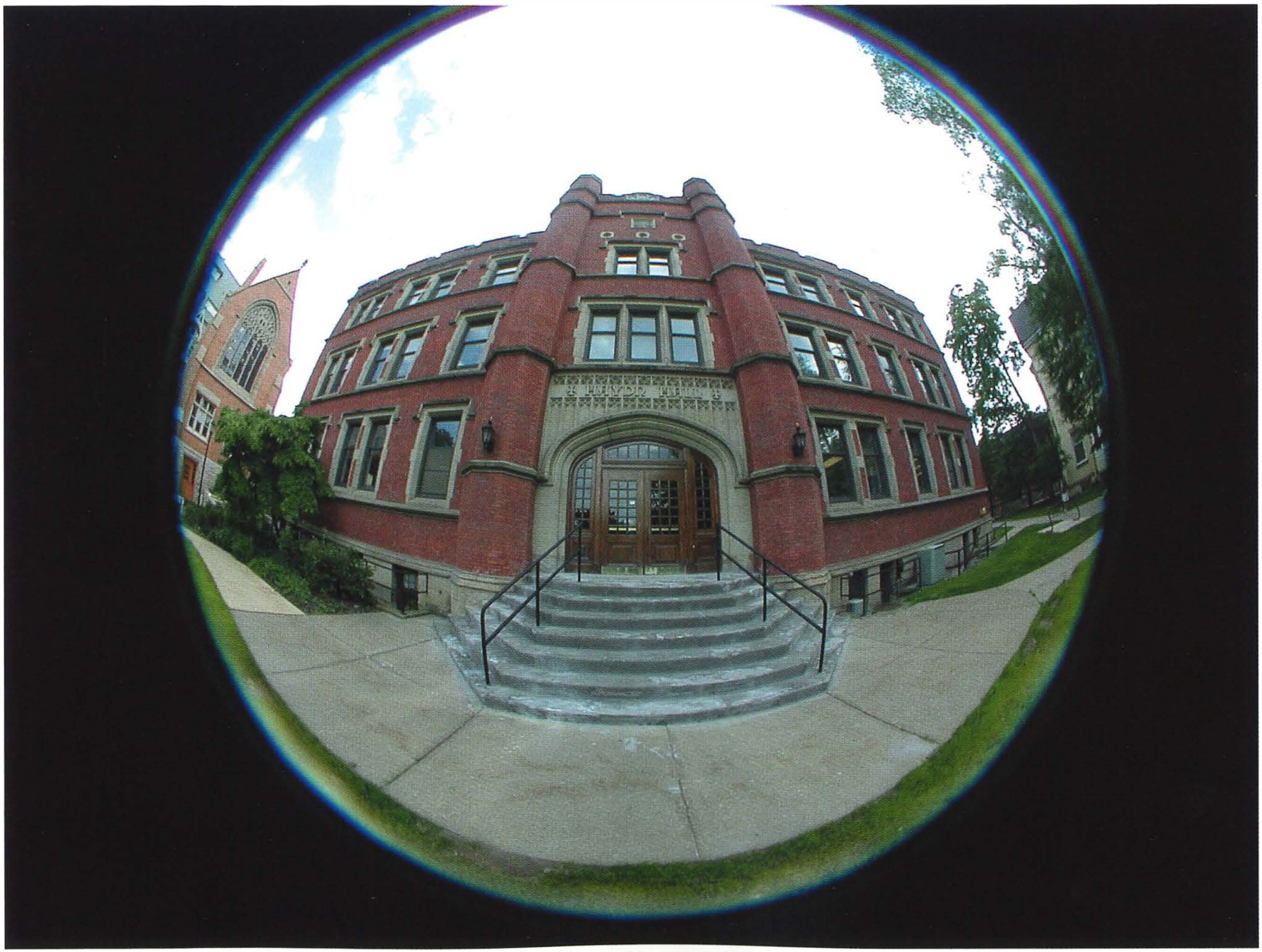


***Figure B** - 8mm - 183° FOV*

The camera has 2 built-in modes for the fisheye converter lens. In Fisheye Mode 1 it produces a circular spot on the rectangular image with a 183° field of view. In Fisheye Mode 2 it becomes a full frame fisheye and fills the entire rectangular image with a 180° diagonal field of view.

I use Fisheye Mode 2 (full frame) for shooting buildings. Though the spot gives a wider field of view there are many reasons not to use it. First, Helmut Dersch's PanoTools plug-ins only allows for fisheye lenses with a field of view of less than 160° (previous versions did not have this restriction, it was removed to avoid an alleged patent infringement). Second, to work with the spot fisheye image it must first be isolated (cropped) which can be tricky since every camera places the spot in a different location. Most importantly the light and color falloff from a spot fisheye is pretty extreme, and as you will see though it geometrically solves the problem it doesn't photographically pick up enough information to accurately represent the desired scene.

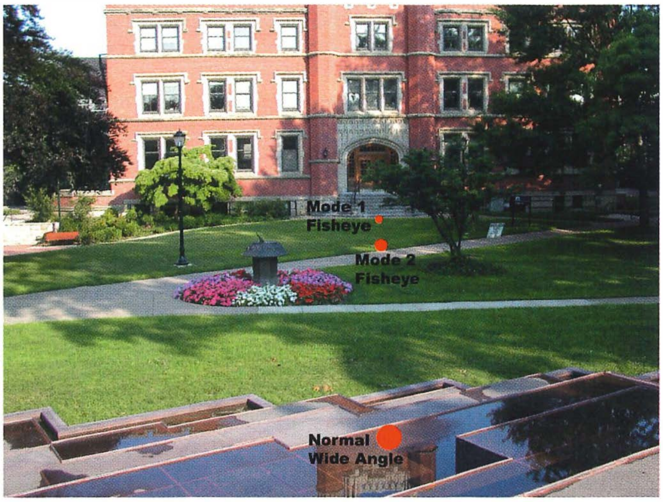
Software: Helmut Dersch's PanoTools

The core tool used in any digital image correction is Adobe Photoshop. Any version of Photoshop (4.0 and higher) will work for this type of image correction. In addition to Photoshop we need to use Helmut Dersch's Free PanoTools plug-ins. These plug-ins are essential for anyone doing work with photographic correction and work on both Windows and Macintosh platforms. After you download PanoTools from http://www.fh-furtwangen.de/~dersch/ and install the Photoshop plug-in. (Don't forget to put the pano12.dll or pano12.lib (Windows and Macintosh respectively) into your system folder) you will now have 2 new filter categories: Pan Controls and Panorama Tools with several filters in each. This tutorial will use the Panorama Tools filter Adjust to un-distort and align the images.

I highly recommend exploring the power of these tools. Helmut Dersch is truly a mathematical genius and these tools are magical.

Shooting

When I go out to shoot a building I normally don't bring a tripod. The ground is rarely level with the building and to get the entire building into my field of view I usually have to tilt the camera up, so I get in good practice of just handholding the camera. As you will see correcting the rotation of the camera will not be that difficult so I can



Location of building shots

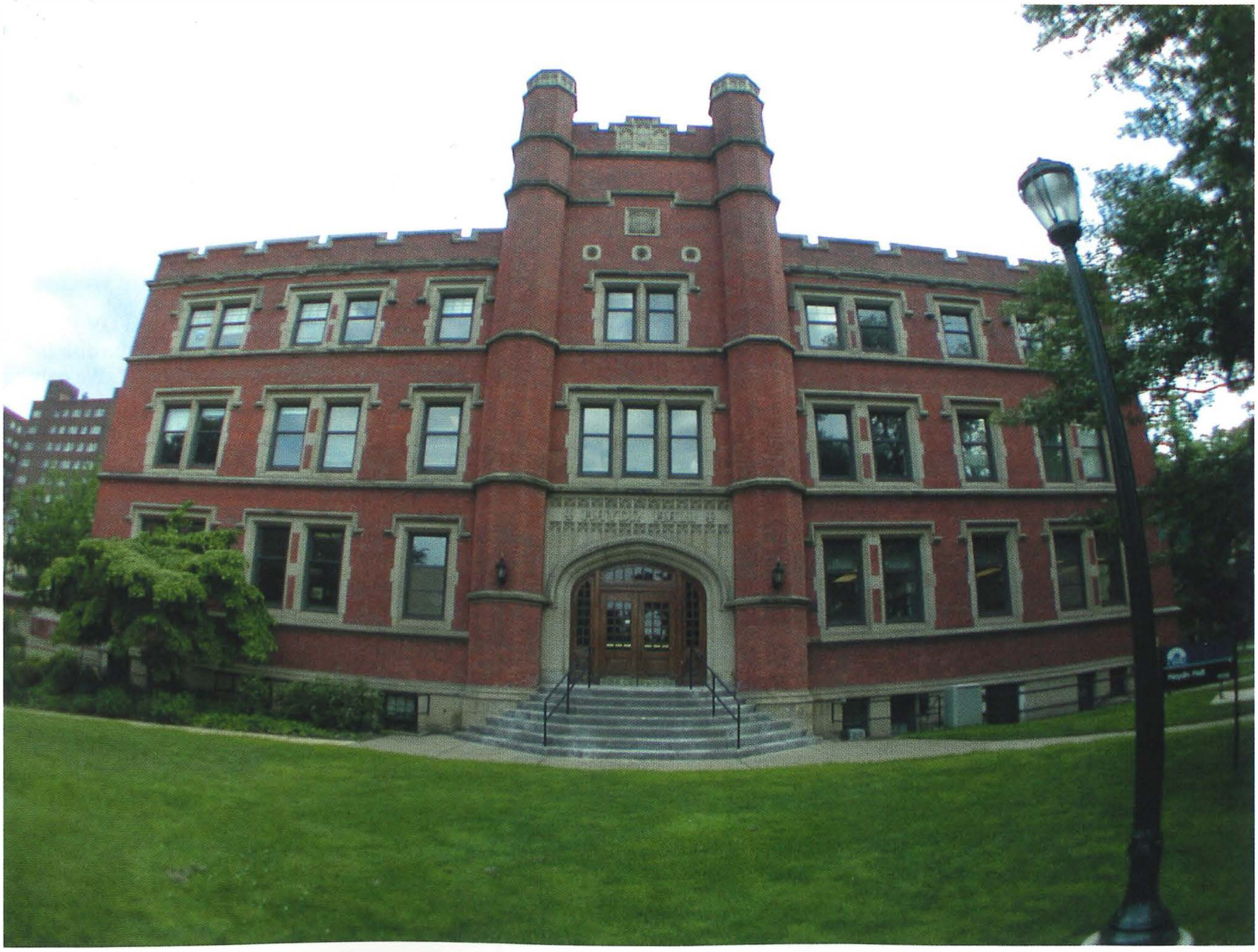


***Figure C** - 8mm - 105° FOV*

focus on just getting a clean shot.

In **Figure A** I used the widest built in lens setting of the camera. You can clearly see the problems; trees cover up most of the right side of the building and the door is obstructed by more trees. What you can't see from the picture is that to get it all into view I had to stand in a fountain (which helped in one way because it made me higher).

In **Figure C** I used the full frame fisheye, which allowed me to move much closer and eliminated the key obstructions in front of the doors and most of the windows. If I wanted to eliminate all of the obstructions I probably would have to move in close and take several shots of the building and assemble them together (but that is a different tutorial).

In **Figure B** I used the spot fisheye and you can see that everything blocking the building is gone (but so is some of the information of the building).

Remember. Don't be afraid to pan the camera up (even if you are using a regular rectilinear lens).

Note the differences in lighting and color in Figures A, B, and C. The lighting did NOT change during the few minutes that I took these pictures. You can still see some of the blue sky in Figures B and C, and the reflections on the windows in all three images are identical. The light and color falloff is another distortion that occurs when using a wide-angle lens (the light has further to travel in the lens to get to the film/CCD). The wider the angle the more the falloff, which is again why a full frame fisheye is more useful than a spot. I also recommend taking several shots without the wide-angle lens so you have reference when you go back to color correct. (These shots are also useful as geometrical reference. You don't want to try to correct linear distortions only to later find out that the building is actually curved that way.)

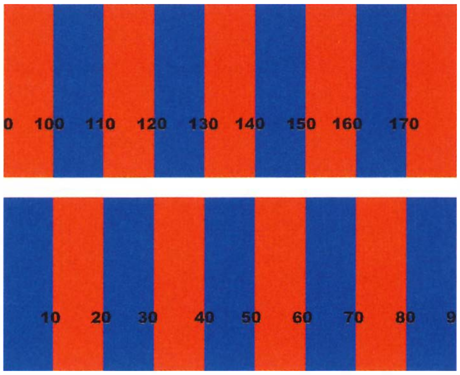
Jared's FOV Finder

In order to correct the linear distortion created by a fisheye or wide-angle lens you must first establish the horizontal field of view of the lens. Some lenses come with a spec sheet that indicates the field of view of the lens (or you can look it up on the manufacturer's website). If you know the focal length you can calculate the field at **http://www.panoguide.com/reference/lens_calculator.html**. For digital cameras or custom lenses (like a wide angle zoom with multiple settings) I prefer to get a good estimate by using my homemade FOV finder.

To make a FOV Finder you will need the following: A 12" circular disk made of wood or Styrofoam (found at a craft or flower store), FOV.JPG (downloadable from my website in the links section), a ruler, scissors, and some masking tape.

Measure and mark the center of the disk.

Print out FOV.JPG at actual size (do not scale or resize it). Cut the printout in half, leave the white space at the bottom of each strip but trim the white space off the right of each strip. Tape the 2 halves together as carefully as possible and make one long strip. Tape the strip to the edge of the disk with the colors and number facing into the disk.



FOV.jpg

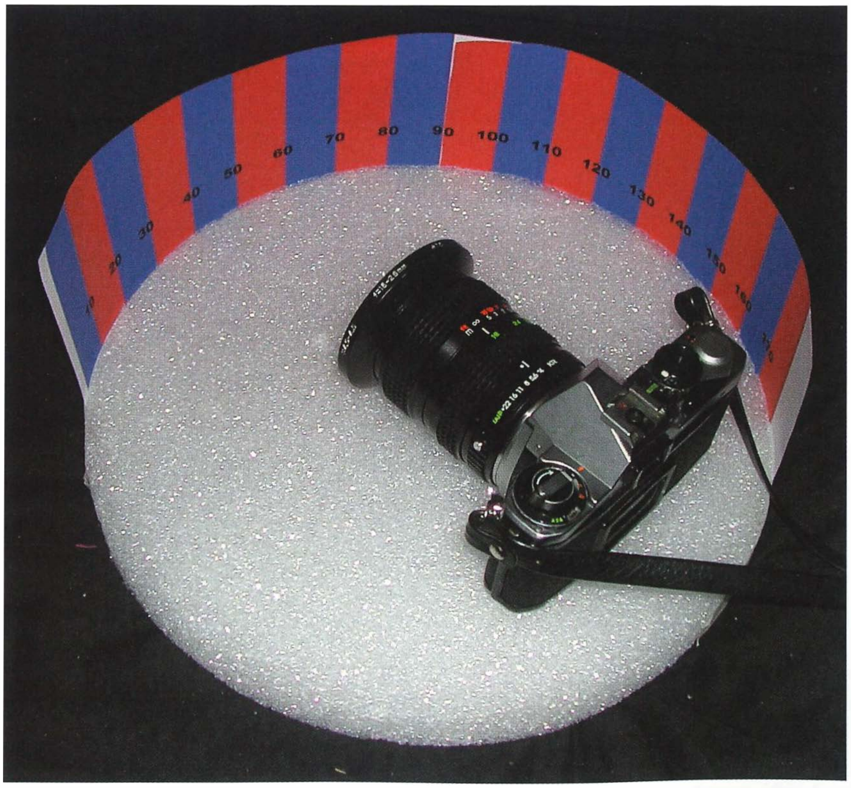
To measure the field of view of your lens you must place the camera so that the nodal point of the lens is over the mark in the center of the disk. Generally the nodal point is located towards the center of the lens barrel and not the camera body (The nodal point is the intersection of the optical axis and the lens' primary plane and you rotate around it you avoid parallax errors). Looking through the viewfinder rotate the camera so you can see just a bit of the white edge on the right side of the camera and the FOV Finder. Focus as close as you can and take a picture. Focus should not be too much of an issue since wide-angle lenses have an extreme depth of field causing most items to be in focus.

FOV Finder Precision

Just what is the precision of the FOV Finder? Well let's do the math. The circumference of the disk is the diameter*PI or 12"*3.14159 = 37.69908". We only calibrate half of the disk so we need a calibration strip that is 18.84954". The calibration strip gets printed at 150 dpi so the strip is 150dpi*18.84954"=2827.431 pixels. We round this down to 2827 pixels. There are 18 - 10° calibration strips in the 180° half circle. 2827 pixels / 18 strips =157.0556 pixels per strip. We round this down to 157 pixels. 157 pixels * 18 strips = 2826 pixels which is 1.431 pixels, 0.00954", or 0.0506% from perfection.

Accuracy? That's an entirely different story and depends on the precision of the printer and the Styrofoam disk. Additionally, the assembly and proper use of the FOV Finder are both subject to user error. Overall, the FOV Finder is a good way of estimating the field of view of most lenses.

Using the FOV finder we can see that the Nikon's built in widest angle setting is a little over 50°, the full frame fisheye (Mode 2) is about 105°, and the spot fisheye (Mode 1) is over 180°. Although these results are only approximate it gives you a good start for fine-tuning the precise field of view in PanoTools.



Jared's FOV Finder

FOV of Nikon Wide-Angle

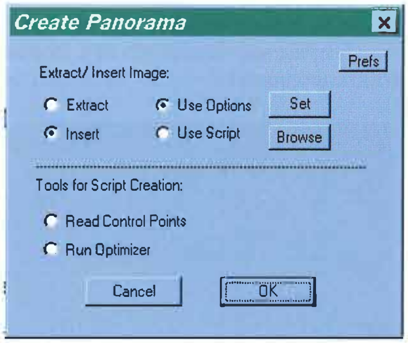
Correcting the Image

The process of correcting an image is an iterative one. You repeatedly tweak each parameter and undo the transformation until each parameter is set. There are 4 main parameters to set: HFOV (Horizontal field of view), Pitch (rotation top to bottom), Yaw (rotation right to left), and Roll (rotation clockwise and counterclockwise).

You need to be sure to undo between each transformation to avoid generational losses. You only want to perform one final transformation.

Load your image into Photoshop and select Filter > Panorama Tools > Adjust. Select the radio items for Insert and Use Options and then click Set.

In Panorama Tools the top half of the Create Panorama Options dialogue box is the source image and labeled Image and the output image is the lower half of the dialogue box labeled Panorama.


Create Panorama

Extract/ Insert Image:

Prefs

○ Extract ● Use Options Set

● Insert ○ Use Script Browse

Tools for Script Creation:

○ Read Control Points

○ Run Optimizer

Cancel OK

Panorama Tools - Adjust Dialog Box

Basic Parameters

Set the Panorama: Format: to Rectilinear and set the Width and Height to about half the size of the image (this speeds up the processing time during the parameter setting stage).

Horizontal Field of View

If you are correcting a rectilinear image like **Figure A**, then you

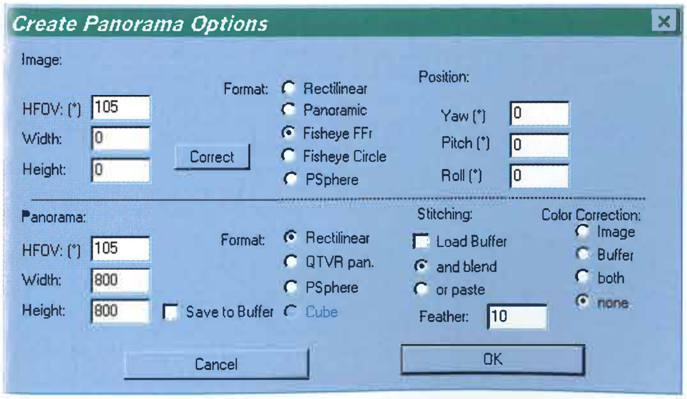

Create Panorama Options

Image:

HFOV: (°) 105

Width: 0

Height: 0

Correct

Format: ○ Rectilinear ○ Panoramic ● Fisheye FF ○ Fisheye Circle ○ PSphere

Position: Yaw (°) 0 Pitch (°) 0 Roll (°) 0

Panorama:

HFOV: (°) 105

Width: 800

Height: 800

☐ Save to Buffer

Format: ● Rectilinear ○ QTVR pan. ○ PSphere ○ Cube

Stitching: ☐ Load Buffer ● and blend ○ or paste

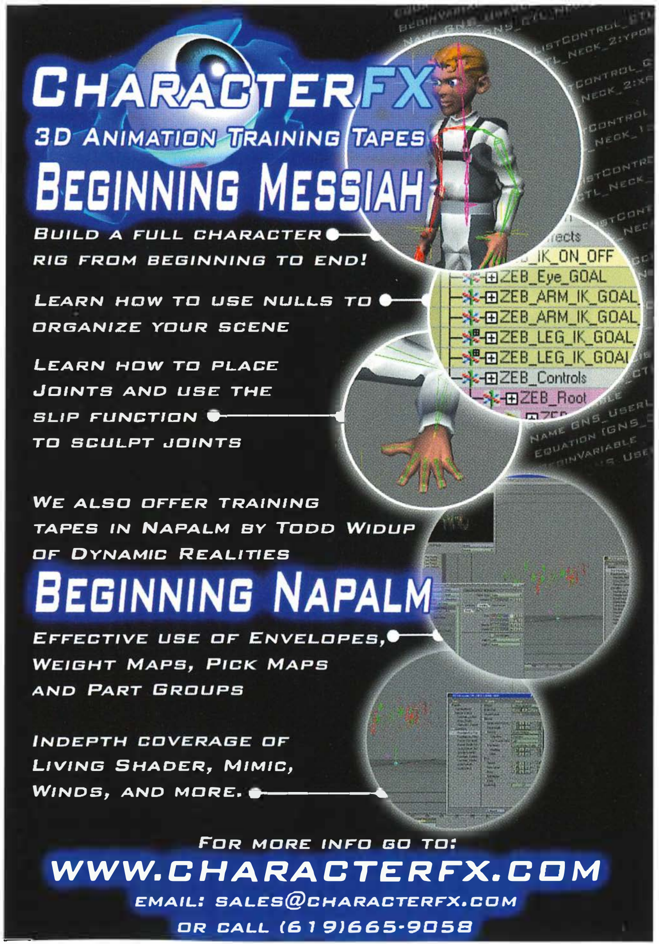
Feather: 10

Color Correction: ○ Image ○ Buffer ○ both ● none

Cancel OK

Panorama Tools - Create Panorama Options Dialog Box

FOV of Nikon Fisheye - Mode 1



CHARACTERFX
3D ANIMATION TRAINING TAPES
BEGINNING MESSIAH

BUILD A FULL CHARACTER RIG FROM BEGINNING TO END!

LEARN HOW TO USE NULLS TO ORGANIZE YOUR SCENE

LEARN HOW TO PLACE JOINTS AND USE THE SLIP FUNCTION TO SCULPT JOINTS

WE ALSO OFFER TRAINING TAPES IN NAPALM BY TODD WIDUP OF DYNAMIC REALITIES

BEGINNING NAPALM

EFFECTIVE USE OF ENVELOPES, WEIGHT MAPS, PICK MAPS AND PART GROUPS

INDEPTH COVERAGE OF LIVING SHADER, MIMIC, WINDS, AND MORE.

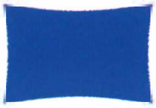
FOR MORE INFO GO TO:
WWW.CHARACTERFX.COM
EMAIL: SALES@CHARACTERFX.COM
OR CALL (619)665-9058

won't need to know or adjust the Horizontal Field of View. Instead set both Horizontal Field of View settings to the same number (it can be anything I usually use 90) and set the Image: Format: to Rectilinear. Do not transform the image yet instead move on to the next parameter.

With a wide-angle lens or full frame fisheye set the Image: Format: to Fisheye FFR and enter the estimated field of view for the lens in both Horizontal Field of View fields. Click OK twice and PanoTools adjusts the image. Visually inspect the output. If the edges of the building appear straight then you have the correct field of view settings. If the edges bow outward you need to increase the HFOV. If the edges bow inward then you need to decrease the HFOV. Undo the transformation, adjust the HFOV settings, and repeat the transformation and undo process until the edges appear straight.

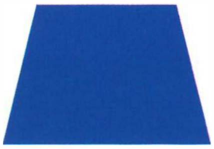
HFOV Adjustments

Lines Bow Out
Increase HFOV

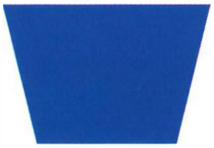


Lines Bow In
Decrease HFOV

Pitch Adjustments



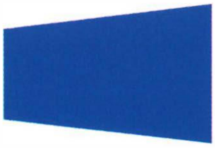
Push Bottom / Pull Top
Increase Pitch



Push Top / Pull Bottom
Decrease Pitch

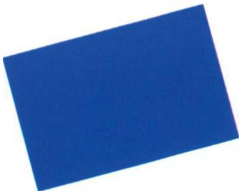
Yaw Adjustments

Push Left / Pull Right
Increase Yaw

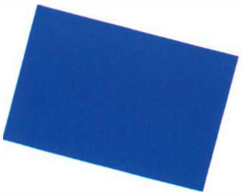


Push Right / Pull Left
Decrease Yaw

Roll Adjustments



Rotate Clockwise
Increase Roll



Rotate Counter-Clockwise
Decrease Roll

Pitch

Visually inspect the previous output. If the bottom edge is wider than the top edge then you were tilting the camera up when you took the picture and need to increase the Pitch setting. If the top edge is wider than the bottom edge then you were tilting the camera down when you took the picture and need to decrease the Pitch setting. Undo the transformation, adjust the Pitch setting, and repeat the transformation and undo process until the width is equal on the top and bottom edge.

Yaw

Visually inspect the previous output. If the left edge is larger than the right edge then you need to increase the Yaw setting. If the right edge is larger than the left edge then you need to decrease the Yaw setting. Undo the transformation, adjust the Yaw setting, and repeat the transformation and undo process until the left edge is equal to the right edge.

Roll

Visually inspect the previous output. If the image needs to be rotated clockwise then you need to increase the Roll setting. If the image needs to be rotated counter-clockwise then you need to decrease the Roll setting. Undo the transformation, adjust the Roll setting, and repeat the transformation and undo process until the image is level.

Size

Visually inspect the previous output. The image should now be straight and rectangular. The top of the image might be cut off so you need to increase the Height setting. Undo the transformation, adjust the Height setting, and repeat the transformation and undo process until you can see the entire image.

The last step is a little tricky. You need to increase the Height and Width settings to maximize the size of the image. Undo the transforma-

SOMETHING YOU SHOULD KNOW...

Ries Graphics ltd. uses the most advanced technology available in the industry to provide graphic and printing services.

AWARD WINNING FULL-COLOR PRINTING:
Magazines, Newsletters, Catalogs, Calendars, Corporate identity pieces, Manuals, Direct mail, many other commercial printing projects

A few of our company's High Tech Features:
- Pioneers in Computer-To-Plate technology
- Experts in electronic pre-press services
- Digital proofing equipment calibrated to our printing presses
- Hi-speed file transfer from anywhere via our FTP site
- Transfer of electronic file color data directly to our eight-color presses
- Ability to print four colors on each side of the sheet simultaneously
- Roll-to-sheet capability to reduce costs and increase productivity
- Web site design and hosting

You are welcome to visit and tour our facility and see our technology in action!

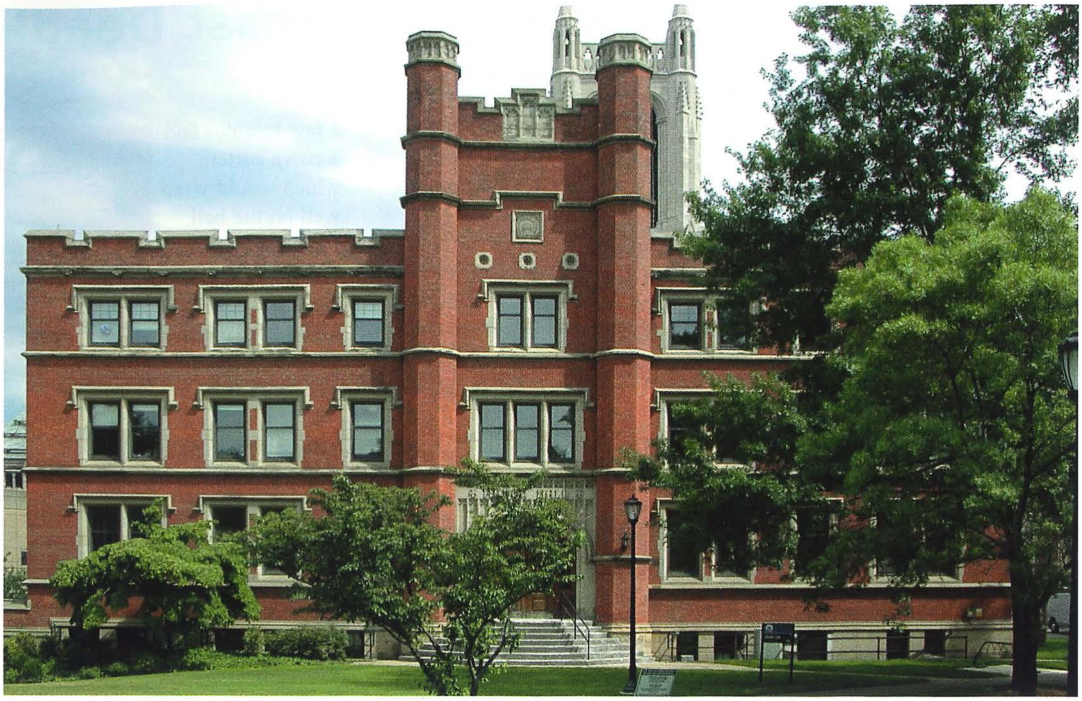
Ries Graphics ltd.
12727 W. Custer Ave., Butler, WI 53007
(262) 781-5720 • FAX (262) 781-1842
www.riesgraphics.com • www.riesnet.com

tion, adjust the Height and Width settings, and repeat the transformation and undo process until the image is as large as possible. Crop the final image and save it as a new file and you are done.

Write down the settings! Nothing is more frustrating than realizing you want to go back and work on a source image and have to run through all of these transformations again.

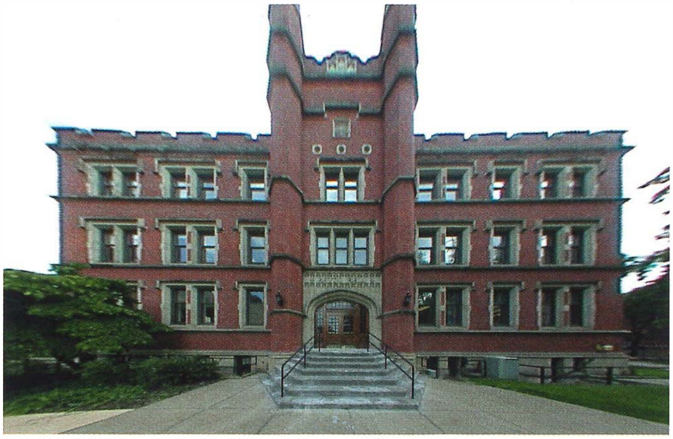
Final Look

Let's take a look at the final versions of our three examples. The finished version of **Figure A** which used the camera at its built in widest setting only required a basic Pitch Adjustment and is heavily obscured by trees and poles.



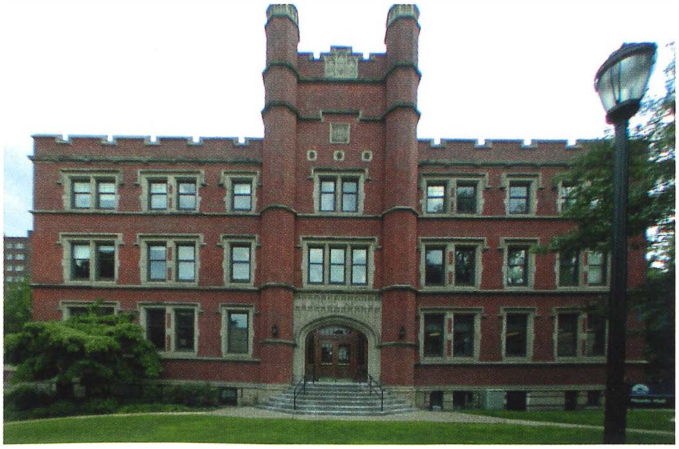
Finished A *- Camera at Widest Image*

The finished version of **Figure B** clearly demonstrates why you should not use a spot fisheye lens. In addition to the light and color falloff, there is image degradation at the edges of the building. Notice the strong exaggeration of perspective on anything not on the flat plane of the face of the building, every window looks 5 feet deep and the top of the building is strangely distorted. Try this with a skyscraper and watch the top almost completely vanish. (Note: This tutorial did not outline the procedure on the conversion of a spot fisheye image, it is just used throughout as a bad example.)



Finished B *- Spot Fisheye Image*

The finished version of **Figure C** has less light and color falloff than with the spot fisheye. Also the only remaining distortions are only on the items that shouldn't be in the picture, such as the lamppost and the sign. Enough of the building is clean enough that I could easily edit out those items. Now all I need to do is shoot the other sides, map them on a box, and render it.



Finished C *- Full Frame Fisheye Image*

Links

Nikon USA: **http://www.nikonusa.com**

Adobe: **http://www.adobe.com**

PanoTools: **http://www.fh-furtwangen.de/~dersch/**

Jared's FOV.JPG: **http://www.jaredjared.com/pano/fov.jpg**

Panoguide Lens Calculator:
http://www.panoguide.com/reference/lens_calculator.html

Jared Bendis is an Instructional Technology Specialist and the Technical Developer for Curricular Virtual Reality at Case Western Reserve University. When he's not inflicting virtual motion sickness on freshmen, he reads email sent to ***jaredjared@earthlink.net****.*

Beating the Broadcast Deadline

by thorn

"Fast, good, or cheap. Pick two."

Just about every 3D artist has said that at some point or another when faced with a project. However, in broadcast animation - more often than not, you don't have much choice except to deliver on all three counts. The very nature of broadcast usually means "fast" is required. Deadlines of "by tomorrow" are not uncommon. Because the project is likely to be seen by hundreds of thousands of people, you want it to be good. And if the client is your employer, the budget may be non-existent; you may have to use only the tools and resources already in-house.

Cinema 4D XL performs quite well in such situations, due to some inherent qualities of the application which can greatly reduce the time and effort needed to produce a good broadcast animation. The non-destructive nature of the object manager, powerful instancing, and the legendary speed of the raytracer make creation and modification much easier on the artist. With XL7, Maxon has added the power of multi pass rendering... which can be a lifesaver when dealing with animations in post.

The key to using C4D XL is a broadcast environment is taking full advantage of its strengths... some are obvious, and some aren't commonly realized for the power they offer. This article will help you get the most out of the application - and hopefully allow you to deliver on all three counts - fast, good, and cheap.

So you're sitting at your desk at 9am, drinking your latte and reading the latest gossip on Postforum when a producer walks in the door and says something along the lines of, "So - ready to have some fun? I need a short image spot for the basketball special we're running tomorrow. I'm thinking... maybe a basketball floating above a pedestal of some kind. I'm using some techno music, so make it kind of sci-fi looking. Oh, did I mention I need it tomorrow?"

And thus our day begins.

The first thing we need to do is quickly plan out the basics of the shot. We'll obviously need a basketball. The sci-fi twist is probably going to involve some polygonal modeling with poly extrusions, bevels... and we'll need a few "props" in the scene to keep it interesting. However, we won't have time to do a lot of objects with detailed modeling. So we'll have to make sure the objects we can create in the time allotted hold the scene together. Good use of texturing will help here. Because it's going to be a rather short sequence (8 seconds in this case), we'll probably only use one camera cut. This means we'll have to use something else to keep things from looking too static... perhaps some particle action or lighting effects will work. Because the shot is related to a specific college basketball team promotion, it will be a good idea to use a color palette which is consistent with the team's colors. And finally, we'll need to keep the effects as lean as possible in order to have the scene rendered overnight.

So, let's start with the modeling.

The basketball itself was fairly easy to create - after all, it's just a sphere. The texturing is the part that actually took the effort. Because I'm using spherical mapping, it's relatively easy to determine the image size that's needed for an optimal spherical wrap: the width should be twice the size of the height. So, I launched Adobe Illustrator and within a few minutes had a curve pattern which would wrap well on the ball. Then the AI file was imported into Photoshop, where I also used a quick plug-in operation to give the ball a basic bump map. The color map (**Figure 1**) and bump map (**Figure 2**) were saved separately, and each imported into a new XL material. Phong mapping was used for the illumination type, with a rather subdued specular curve. I also used SAT mapping for the image maps, to ensure the sharpest texture possible while nearly eliminating any possibility of texture flicker.

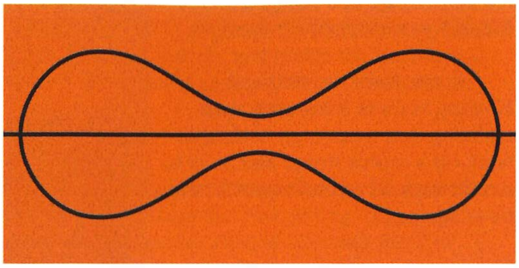


Figure 1: *The basketball color map.*

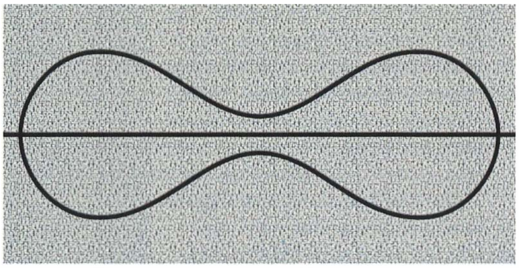


Figure 2: *The basketball bump map.*

Upon applying the textures to the basketball, I noticed a problem: the vertical stripe on the ball was smearing due to the spherical distortion. Now we could get creative in Photoshop with the Polar Offset filter, but time was the enemy and a faster and easier solution was needed. I created a new 100% black material, and planar mapped it onto the basketball, scaling it such that it would be just the right width along the vertical axis. (**Figure 3**) Problem solved. Now the purists among us will point out that the vertical stripe isn't accounted for in the stripe bump mapping. This is true, but when the ball is rotating about the scene while the camera moves as well, it isn't noticeable. I saved the ball object, and started to plan out the "pedestal set."

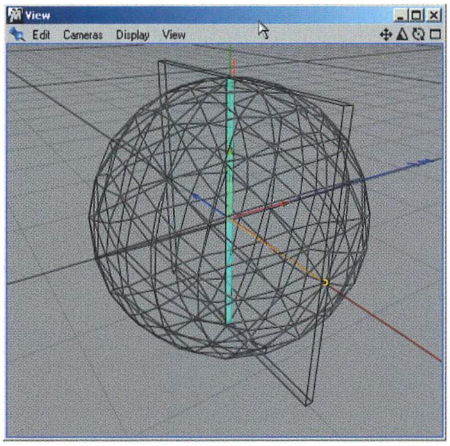


Figure 3: *Planar mapping the basketball's vertical stripe.*

Instead of going with the age-old approach of a ball rotating above some cylindrical obelisk, I decided it might be more interesting if the ball emerged from the floor in a rather impressive fashion and then came to its final position. The sci-fi angle immediately reminded me of those spiraling circular doors, so I decided that might be a good scene element to start with. The basic premise is that the door is made up of a group of interlocking, curving sections. Instead of trying to model the whole thing in XL and manually adjust the sizing for the rotational duplication, I decided Illustrator would be the best starting point. I drew out a circle, and then a curved section running from the center of the circle out past the border. Then using the rotate/copy function, I offset the curve 15 degree. I then ran the Intersection filter in Illustrator to split the segments apart, deleted the unnecessary pieces, and then used the Unite filter to get a complete, closed spline shape for the door piece. (**Figure 4**) Finally, the shape was saved as an AI file.

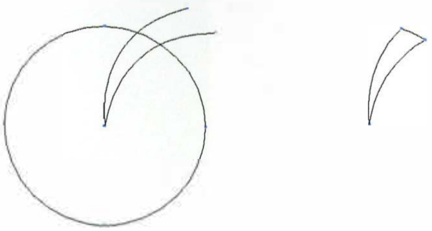


Figure 4: *Left: the splines before the intersection operation. Right: the final result.*

I imported the AI file into XL, and used the Extrude NURBS object to create the full geometry. The capping included a 1-segment bevel, with Constrain Contour turned on to keep all geometry with the boundaries of the original spline. Because the shape of the object was critical in all the duplicates fitting together, I decided to convert the object to polygonal geometry so that I wouldn't risk accidentally editing the spline again. (**Figure 5**) Now it was time to make the duplicates. The easiest way of doing this is to duplicate the object around the Y axis, so I changed to the object axis tool and turned on point snapping. I snapped the axis of the object to the point at the very tip of the tooth, changed back to the model tool, and set the object coordinates to world 0/0/0. Finally, I created a null object at the 0/0/0 position and made the object a child... this will be important when we instance and animate the object.

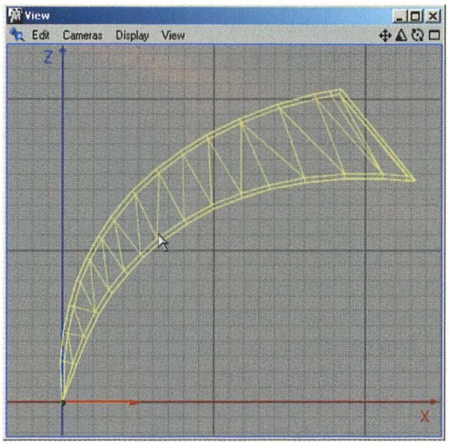


Figure 5: *The master door segment object.*

Selecting the null parent of the door piece, I chose Duplicate from the Functions menu and entered "24" for the number of copies (360 degrees of a circle, divided by 15 degrees from the original Illustrator template = 24 door segments) and checked "Create Instances." I zeroed out all the fields, except for H (heading, or X axis) which was given 360. (**Figure 6**) Now, you may be wondering why I've created 24 copies instead of 23... the answer is simple. When XL creates duplicates on a 360 degree rotational axis, it creates the last duplicate in the SAME position as the original object. So after the copies are made, I went into the object group and deleted #24. I then grouped the instances and the original under a common null parent. (**Figure 7**) With the basics of the door completed, time to move on to the pedestal itself.

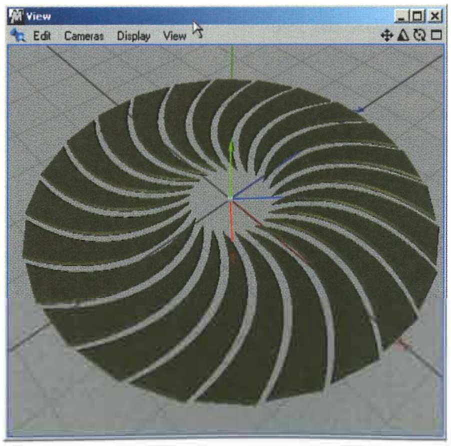
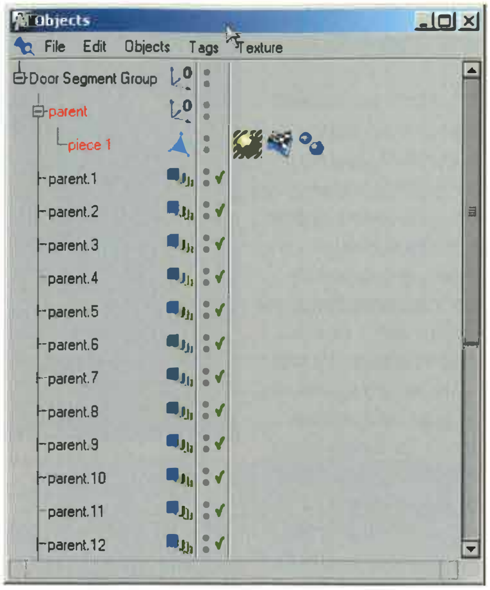


Figure 6: *The 24 door segments, arranged in a circular pattern by the Duplicate function.*



Door Segment Group
  parent
    piece 1
  parent.1
  parent.2
  parent.3
  parent.4
  parent.5
  parent.6
  parent.7
  parent.8
  parent.9
  parent.10
  parent.11
  parent.12

Figure 7: *The master object, with the instances and proper hierarchy.*

The pedestal/stage object is really quite simple. I started with a tube primitive object, sized to fit the door opening. After performing some poly extrusions, inner extrusions, and beveling I had a shape I was happy with. This could easily be done with a lathe object and a spline, but I tend to prefer the free-form modeling approach when it comes to mechanical-type things. The important thing to remember when creating an object like this is that you want to be sure the object will have sufficient surface detail in order to look interesting - basically, you want a lot of nice beveling to catch specular highlights. If you find the object is being overly affected by the smoothing tag, you can double-click the Smoothing Tag in the object manager and reduce the angle (I generally find that using an angle limit of 35 degrees is sufficient.) However, sometimes you may find the object's breaking angles are so shallow that you either get too much smoothing, or not enough. In this case, you'll want to select the ring of polys which are the angle of the bevel, and choose the Disconnect/Preserve Groups function. This effectively doubles the edge points, and disconnects the polys from the neighbors while keeping the object itself in one piece. With the edge points doubled, the smoothing tag will overlook this area and you'll get a nice, non-smoothing breaking bevel. (**Figure 8**)

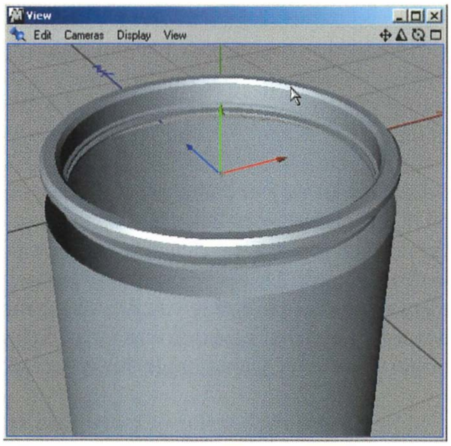


Figure 8: *The basic tube object, complete with bevels and poly extrusions.*

The inside of the tube looked rather dull, being completely smooth. So, I decided some type of detail was needed there. I tried a bump map or two, but found it was still looking far too regular and boring. After a few different attempts, I finally decided on a basic "nurnie" approach. For just a few polys, it's fairly simple to do some selection and extrusion... but I needed a ton of them. Fortunately, fate answered in the form of a plug-in called "Structure" by Ha Mang-Ky which completely automates such tasks. (Hopefully the plug-in will be shipping by the time of this publication, if not soon after.) With the main tube complete, the basic set was almost done. (**Figure 9**)

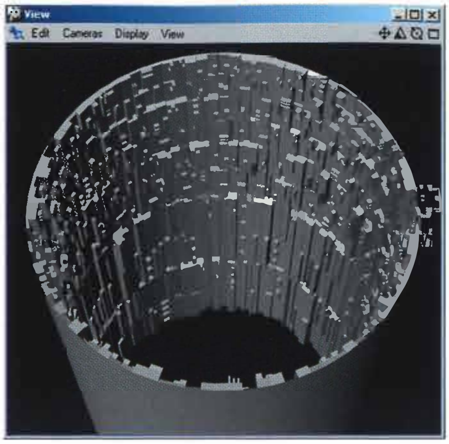


Figure 9: The inner detail of the tube, created with the Structure plug-in.

There are several ways I could have approached the surrounding rectangular area... Booleans appear to be the obvious choice, but generally they create artifacts which I don't care for. I decided instead to select a row of polys around the tube, and extrude them a far distance out on the XZ plane. Once the extrusion was complete, I turned on point-to-grid snapping and snapped the points into a square shape at the border. Another round of poly selection and beveling added some surface detail to the object. Finally, I made 2 selection sets of polys on the surface of the "stage", for future use in texturing. (**Figure 10**)

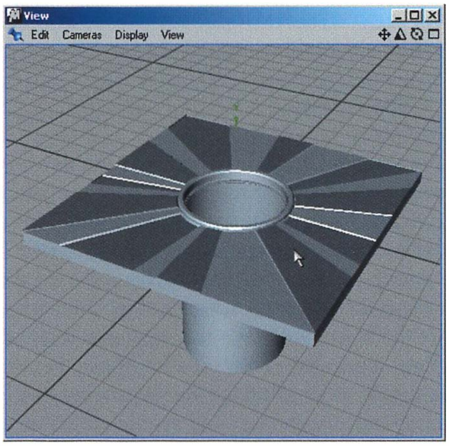


Figure 10: The final stage object, with beveling for extra detail.

The edge of the stage was looking a bit bare, and after a bit of thinking I decided it was because there was no "border" of any type separating the square from... well, the empty space. In addition, the area around the doorway still seemed too uninteresting. Without any vertical detail, it looked too flat. The path I decided to go with was to create some irregular piping structures around the door, as well along the borders of the stage... almost like some type of guardrail.

I find the best way of doing this is to use a combination of sweep NURBS and instances. The great thing about doing it this way is that the creation process is completely non-destructive. Once the basic shape is done, you can further adjust the curve interpolations via the object manager in order to ensure you don't have more polygons than are needed. The use of instances is obvious. You can modify only the original, and the changes ripple down through all the instanced copies.

I created a circle spline primitive, and scaled it such that it was just larger than the door opening. I made 4 copies of it (3 to model with, 1 as a backup). Then, 4 of the spline primitives were converted to point spline objects. After entering point edit mode, I selected the knife tool and made some cuts in the circles, and deleted segments along the way so that I finished with 4 stacks of spline curves. I placed each spline into a sweep NURBS object, and using a 5 unit diameter circle cross section the pipes were nearly complete. Finally, I instanced each copy a number of times, and offset them vertically to create the set of rails. A similar method was used for the sets of rails on the 3 edges of the stage. The main difference was that I only created one side of the rail, and used a symmetry object to mirror the other half. This made sizing the length of the rail to fit the stage a simple matter. A simple plane with an angled corner was placed below the top pipe section, to add more detail to the borders. Once one side was complete, the 2 pipes and the plane were grouped, duplicated twice as instances, and rotated/positioned into place. (**Figure 11**)

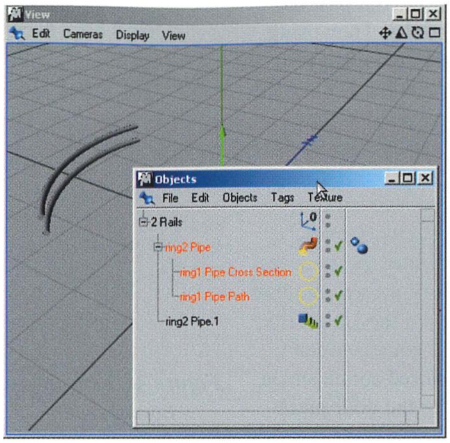


Figure 11: Two of the sweep NURBS rails, with the instanced object hierarchy.

Rails floating in space are no good, so I created a capsule primitive for some support posts. Once again, some detail was needed to catch the viewer's eye. I converted the capsule to a polygonal object, and with a few poly row selections and bevels, it looked much better. Using the duplicate function, I made 7 instances. I turned on spline snapping, and each post was snapped to the end of a pipe with no alignment hassle. Finally, they were grouped in pairs and positioned on the world Y axis as needed. Simple filleted cylinder primitives were used to add "feet" to the pipings along the stage border. Remember, it's always the small details here and there which help keep a scene held together in the eye of the viewer; pipes tend to look strange when they grow out of a flat surface with no fitting. (**Figure 12**)

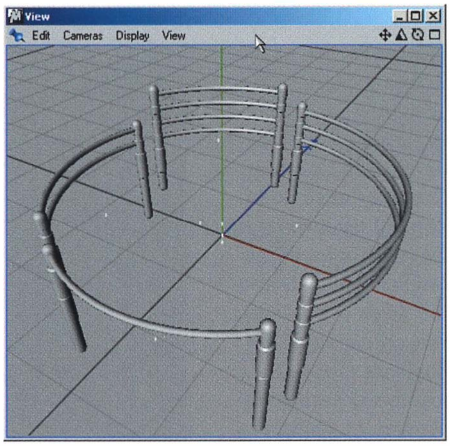


Figure 12: The final rails, with posts in place.

With the modeling done, it was time to move on to the texturing and lighting. The materials are fairly simple in composition, but a lot of time was spent on them to make them work well with the lighting. I knew I had to work with a blue and silver based color palette (due to the team colors), but making the door silver didn't draw the eye enough. Because the camera movement is a basic pan around the scene, I needed to draw the eye where the action was going to take place. After a bit of trial and error, I decided to go with a gold color for the door. (**Figure 13**) The metallic materials all have several things in common. First of all, Blinn shading was used... this works great for metals. Secondly, a turbulence shader was added in the diffusion channel to affect both the

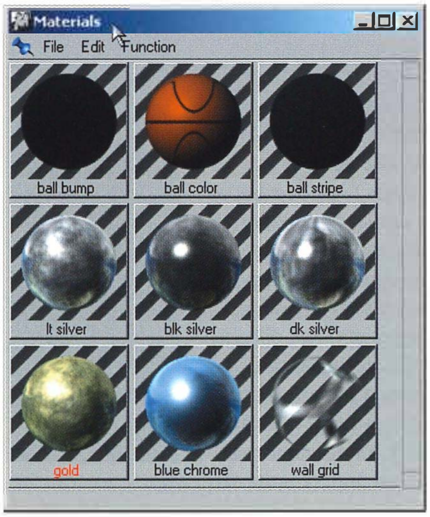
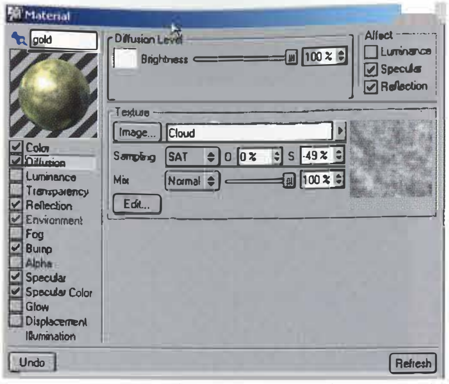


Figure 13: The material manager, with the 9 materials used in the scene.

specular and reflection (note: in XL, the reflection checkbox in the diffusion channel also affects the environment channel). A Fresnel shader was added in the reflection channel, and a suitable bitmap was used for the environment map. Finally, I made liberal use of XL7's new Falloff control in the specular channel to get a hot specular, while not blowing out the edges more than desired. A cloud shader was added into the bump channel of the gold material; combined with the rotational duplication of the instances, this resulted in a stamped metal look I was pleased with. (**Figure 14**) For the outer rail planar walls, I duplicated one of the chrome materials and added a bitmap into the alpha channel. Note: when applying materials to objects which are instanced, you don't need to apply the material everywhere. Simply put it on the main (original) object, and the instances will automatically take the same material and projection settings.



gold

Brightness 100 %

☐ Luminance
☑ Specular
☑ Reflection

Cloud

Sampling SAT O 0 % S -49 %

Mix Normal 100 %

☑ Color
☑ Diffusion
☐ Luminance
☐ Transparency
☑ Reflection
☑ Environment
☐ Fog
☑ Bump
☐ Alpha
☑ Specular
☑ Specular Color
☐ Glow
☐ Displacement
Illumination

Figure 14: *The diffusion channel of the gold material, using the cloud procedural shader.*

I used a fairly standard 3 point light setup for the scene, with some color variation in the lights to add contrast. Using completely white or gray lights is rarely a good idea; it tends to flatten a scene. In this scene, I made the key light slightly blue, while using darker yellow and blue colors for the fills. All the lights were spots, using a target tag in order to quickly position them. 2 final spotlights were added to target the basketball - one as a key, and another illuminating the ball as a back light. Shadows were only enabled for the main scene key light; casting shadows from every light in a scene usually looks odd, and also adds unnecessary time to the rendering.

Time to animate. (**Figure 15**)

The first element I decided to tackle was the door itself... and here's where once again XL's instancing was a great timesaver. Because animation occurs relative to parent space, I was able to animate only the original door segment along an XZ diagonal - and all the instances inherited the position track. To complete the door animation, I recorded a couple of rotation keyframes on the main parent of the door segment group. I ctrl-dragged the door keyframes to create copies, so that the door could close after the ball passed though.

Next up - literally - was the basketball. Probably the easiest part of the scene... a keyframe for the initial position, another for the finish, and some heading rotation along the way to make it more visually appealing.

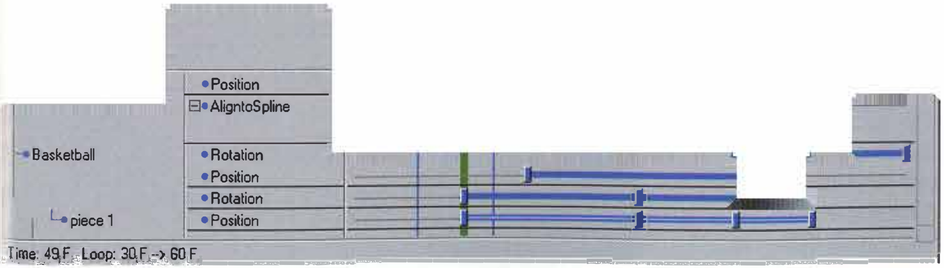


Figure 15: *The timeline with all animation tracks. Note: use of layer assignments helped to clear the "clutter" of non-animating objects.*

The camera animation actually gave me a bit of trouble at first. I started with the standard point-to-point positional keyframing, while rotating the camera along to the way to point in the correct direction. After a few go-rounds with the function curves, it became clear this wasn't the way to go... the camera wasn't tracking the right directions, the motion (although smooth) wasn't what I was looking for either. As often happens, the easiest way is usually staring at you. And in this case, it was using spline-path animation.

I selected b-spline tool and drew a curved path around the door area, naming it "Cam Track". Then, I created a null object near the door center and targeted the camera to the null. Next, I went into the timeline and added an Align to Spline track and created a keyframe using Cam Track spline for the keyframe data. (**Figure 16**) The movement was almost perfect, but at a couple of points in the animation the camera wasn't pointing exactly where I wanted it to. The solution was simple: I gave the camera target a small bit of positional keyframing, so that the camera would track from the door area, to the basketball as needed. Finally, I added ease in/ease out curves to all the appropriate animation tracks via Time Control. (**Figure 17**) The easiest way to create an ease in/out: Select the animation track. Change to the time control window. From the Curves->Time Curves menu, choose Ramp Up. Then select the 2 points on the curve, and select Curves->Tangents->Soft Interpolation. Finally, choose Curves->Tangents->Flat.

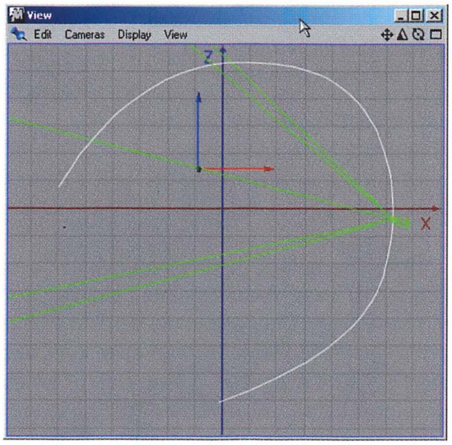


Figure 16: *The scene camera, assigned to a spline path. The camera target is selected in the window.*

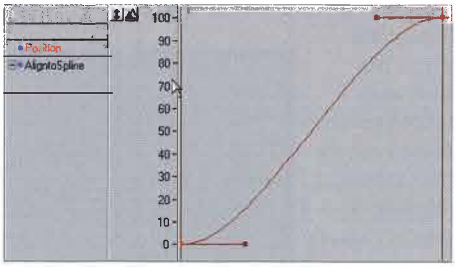
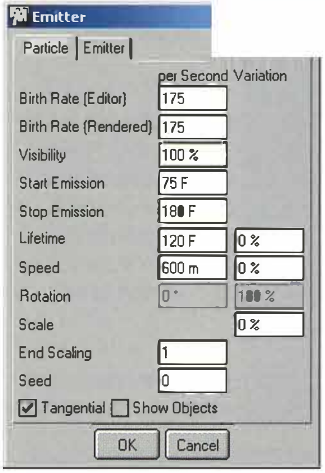


Figure 17: *The ease in/out time control curve for the camera's position track.*

Everything looked complete as this point, but for some reason the scene wasn't attention-grabbing enough. There was no "Hey, something cool is coming!" factor going on. A couple of minutes later, I decided on a volumetric light/particle system combo for the door-open animation.

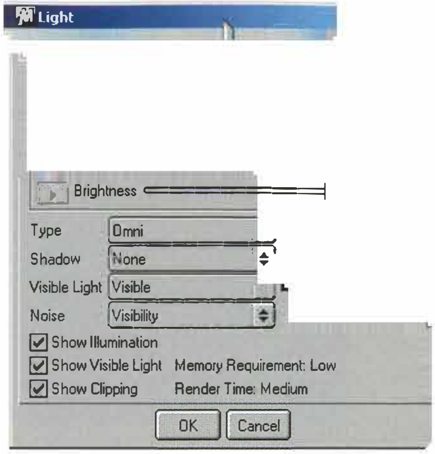
I created an orange volumetric spotlight, and turned on the noise feature as well. It was placed beneath the door, pointing upwards. Note: unless shadows are turned on for a visible light, the light will not behave volumetrically. I was a bit worried that the light was going to add too



	per Second	Variation
Birth Rate (Editor)	175	
Birth Rate (Rendered)	175	
Visibility	100 %	
Start Emission	75 F	
Stop Emission	180 F	
Lifetime	120 F	0 %
Speed	600 m	0 %
Rotation	0 °	100 %
Scale		0 %
End Scaling	1	
Seed	0	

☑ Tangential ☐ Show Objects

Figure 18: *The settings used for the particle emitter.*



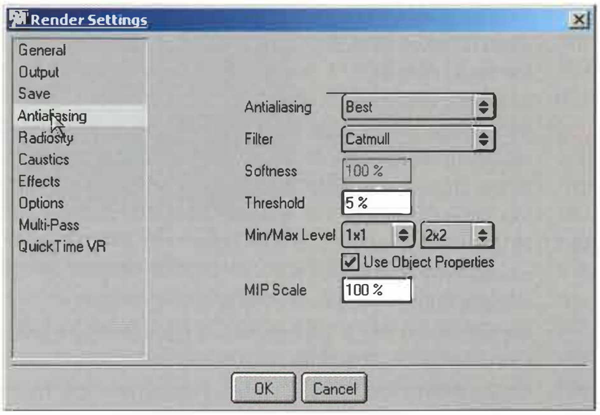
Type Omni
Shadow None
Visible Light Visible
Noise Visibility
☑ Show Illumination
☑ Show Visible Light Memory Requirement: Low
☑ Show Clipping Render Time: Medium

Figure 19: *The main light settings for one of the visible light particles. Note that "No Light Radiation" is checked to save render time.*

much time to the render, but after doing a couple of frame tests it turned out it was only adding a couple of minutes to each frame's render time - still under my goal of 5 minutes per frame.

After the volume light was completed, I set up a particle system (**Figure 18**) using 2 small visible lights as particles (**Figure 19**). I set the start and end time within the particle system to have the particle stream correspond with the door animation, and added a turbulence object to give the particle motion a bit of irregularity. Combined with the volume light, it really took the scene to another level.

So - all was set to go, except for setting a couple of rendering options. I set the AA to Best, and used Catmull for the AA method. The min/max for the scene was set to 1/2, with a threshold of 5%. (**Figure 20**) After a few frame tests, I decided the thin geometry of the rails (combined with their material) wasn't clean enough, so I added a render tag to the rail objects and set the min/max to 2/2. Forcing the AA to level 2 solved the jaggy problems, and the per-frame render time still did not exceed 5 mins/frame. Total render time - right about 14 hours for D1, which was perfect in time for the production deadline.



Antialiasing Best
Filter Catmull
Softness 100 %
Threshold 5 %
Min/Max Level 1x1 2x2
☑ Use Object Properties
MIP Scale 100 %

Figure 20: *The anti-aliasing settings used for the final render.*

When doing animation on a fast deadline, it's always important to do only what is absolutely necessary to make the scene work. Modeling only what is visible, using instances to help cut down on re-modeling entire groups of objects, and making your textures as effective as possible. In most cases, it's usually not about using every tool or feature in the application; it's about making the best visuals with the features you DO use.

-- *thorn@maxon.net*

ELECTRIC RAIN'S SWIFT 3D PLUG-IN FOR LIGHTWAVE

Electric Rain, Inc.
http://www.erain.com
sales@erain.com
888-613-1500
303-543-8230

$295

reviewed by Kyle A. Thatch

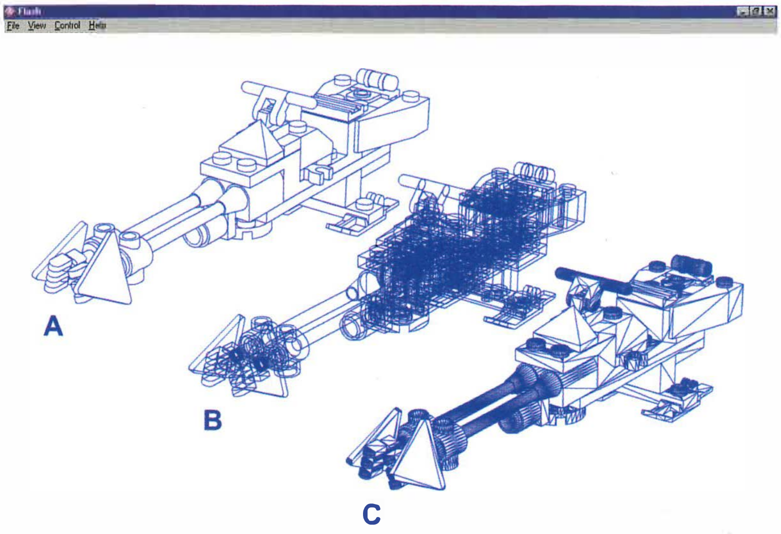


Figure 1

Unless you have been living in a cave without electricity and other modern conveniences, you have probably noticed people using Flash to spice up their websites. Flash is a vector-based program that allows you to do 2D animation on the web. It is possible to do 3D with Flash, however it is not very easy compared to using LightWave 6.5b to create and save 3D animations. Don't despair, though. There are a group of people at a company called Electric Rain that have developed a product call Swift 3D to help us tri-dimensional guys and gals do the unthinkable. Export our third dimensional animations to Flash.

Many of my close friends know that I have a trained pet monkey called Bonzo. Occasionally, I will take Bonzo and chain him to my computer desk and let him learn new software for me. OK, so maybe I am stretching the truth a little, but it would almost be possible to do just that if I had a monkey named Bonzo. Swift 3D has got to be one of the easiest programs I have ever installed and learned to use with LightWave. Swift 3D adds a Tab to the layout portion of LightWave. This is where all of your selections and rendering takes place. No shaders to add to objects, no rendering plug-ins to install and execute. Just a single tab; that's it.

There are four directories to choose from the Swift 3D Tab. The most important directory is the Support directory. Not only does it explain the program so eloquently that my friend Bonzo can understand, but it is packed full of websites showing examples and places to hang out and talk about Swift 3D. Electric Rain has really impressed me with the thoroughness of their support network; other companies could learn a lot about how to support a product from this.

The meat and potatoes of the program are located in the Render Options Tab. Swift 3D offers the widest variety of vector formats I have seen from this type of Plug-in. They include: Macromedia Flash (SWF), Encapsulated Postscript (EPS), Adobe Illustrator (AI), and Scalable Vector Graphics (SVG). You simply tell Swift 3D where you want to save your files using the Save as button, then select what type of file to save. In the case of saving Flash files, you can select which type of Flash files to save. You can choose Flash 3, 4, or 5. A nice feature is the Preview only button. This will render your Animation or File for you to see how it will look and to get a feel for the rendering time, but will not save the Animation or File. Just uncheck the button to save your work.

When it come to rendering, there are two tabs to choose from: Fills and Edges. You may also use these two Tabs in conjunction with each other. Take a look at **Figure 1**. 1A shows a simple Lego block rendered using outlines only. 1B shows the same block with Include Hidden Edges turned on. Finally, 1C shows Entire Mesh Outlines. I do not think most LightWave users will use this option, but it is included mainly because of the fact that Swift 3D was originally developed for 3D Studio Max. You can change the outline thickness and color as well as the level of detail the lines receive from this menu.

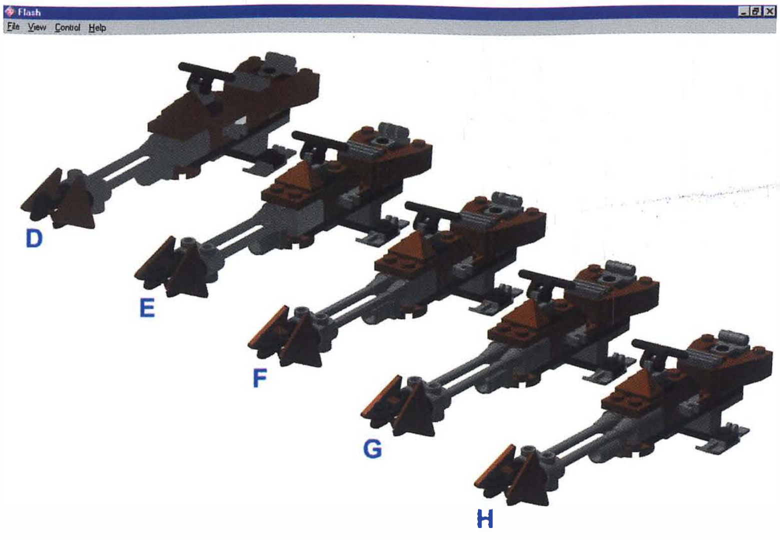


Figure 2

Do you like Cel Shader? Then you will love the Fills tab. With 5 cartoon levels to choose from, it is bound to make any Animé wannabe proud. Looking at **Figure 2**, you can get a good idea of the different rendering levels available: (A) Cartoon Single Color Fill, (B) Cartoon Average Color Fill, (C) Cartoon Two Color Fill, (D) Cartoon Four Color Fill, (E) Cartoon Full Color Fill. Hey, what about Gradient Shading you ask? No problem. There are two of those to choose from also. Need shadows and specular? What about the ability to cast a self, plane or object casting shadow? No problemo muchachos. I am telling you, this plug-in will do you and your monkey proud. Take a look at **Figure 3**. It is hard to get closer to a directly rendered output than that.

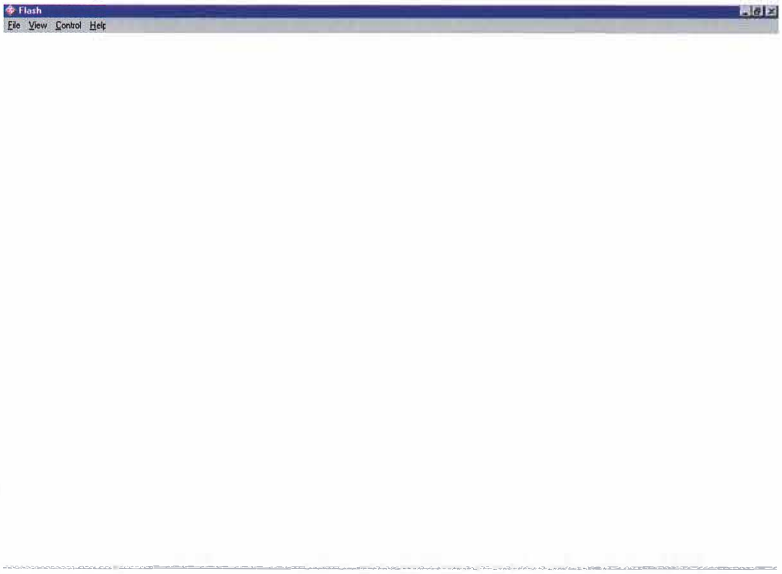


Figure 3

Once you make your selections from the Render Tab, you can select to render a single frame or an entire animation using the Render File or Render Scene buttons. That is pretty much it. If you read through the outstanding manual and fiddle around with the rendering options, you will be outputting to Flash or Illustrator almost as fast as you can install Swift 3D and load a scene. OK...OK what is the bottom line, the pros and cons, you say?

Pros:
- Easy to Install and Learn (Monkey friendly)
- Fast Render Speeds
- Exports a wide variety of Files
- Outstanding Support

Cons:
- When rendering to Illustrator must use (EPS) for Illustrator 8 or below
- Must first triple Models polygons (will not read Quadrangles at this time)
- Does not support object dissolves or multiple lights for shadowing

The bottom line is simple. If you need to create cool, 3D content for the web using Flash or need to Illustrate a 3D object for print, you need Swift 3D. Chain your Monkey to the desk...you'll be glad you did.

A First of all, you must be running LightWave on an operating system that actually recognizes multiple processors. Windows NT or Windows 2000 both support multiprocessor systems but the more conventional operating systems, like Win98, Win98SE, and WinME do not recognize more than one processor. Another excellent multi-processor operating system is Linux, but unfortunately there is not a Linux version of LightWave available yet. (hint hint, NewTek!)

Once you have LightWave installed on a multiprocessor compliant operating system, you need to tell the program to use more than 1 "thread" so that it allocates more processing power to the task of rendering. You can do this in the Rendering Options panel, under the Rendering tab. You might think that setting the Multithreading option to 2 Threads will be sufficient to boost render speed but you will get better results by assigning at least 4 Threads to LightWave's render engine. If you have a very fast processor, something in the 1.5ghz-or-above range, then you may even be able to get away with assigning 8 Threads to LightWave, but don't expect to be able to use the computer for anything else while rendering with this many threads. In fact, if you do assign 8 Threads to LightWave's render engine, then you should make sure that any unimportant applications are not running in the background, especially stuff like screen savers or e-mail readers or anything that can take away system resources from LightWave.

Another way to get the most out of a multi-processor system is to have one CPU rendering an animation with a Multithreading setting of 1 or 2 Threads and then run another copy of Layout or Modeler using the 2nd CPU. This allows one CPU to knock out a finished scene that is ready to render while you use the other CPU to set up another scene or model objects. The advantage to doing this is that you can be rendering out test animations of completed scenes while setting up more scenes to render. This would be the ideal way of using all that processor power during the day when you are feeling productive and then at night, when you're sleeping or whatever, you can have both processors crunching out rendered frames with a multithread

A multi-processor system can be a great way to increase productivity, but always keep in mind that 2 processors does not mean your system will do everything twice as fast. You will see a nice increase in render speed with a dual processor system, but it will (almost) never be a 100% increase.

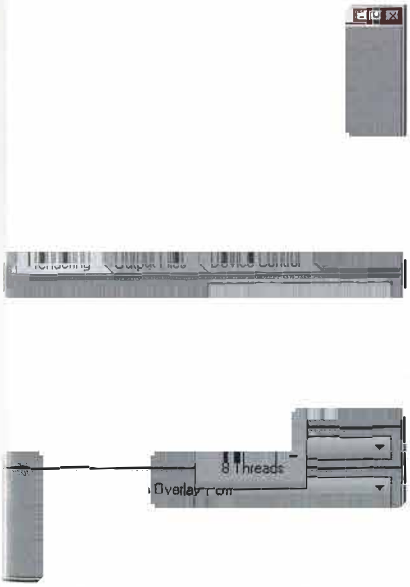


Figure 3: The Rendering options panel with the Multithread setting highlighted.

Q Can you please explain how adding a map or gradient to the diffusity channel would be helpful in surfacing an object? Wouldn't I get the same results if I just painted darkness details into a texture map applied to the surface's color channel?

A The advantage of using a diffusity map in conjunction with a color map is that it just gives you a lot more control over the final appearance of your surface. For instance, if you created a car object and made it look brand new sitting in a showroom, then there wouldn't be any real need to apply a diffusity map because you want the car to look new and flawless. However, if you put that same car in the desert, racing across a dusty landscape, then having the surface of the car looking all shiny and unblemished would not be very realistic. You could certainly go ahead and create a color map that had a dirty look to it, but this would just be a pain to manipulate each time you wanted to adjust the car's appearance. By applying a "dirt" map to the diffusity channel, you can easily control the level of "dirtiness" on the car; when it comes time to give the car a good washing, it's a simple matter to fade out the diffusity map and give your car that shiny new appearance again.

As a rule of thumb, it is best to layer maps on multiple channels and attributes of an object's surface. Trying to do everything with one of the surface's attributes or with one texture map is just inefficient and severely limits flexibility in surfacing your object down the road. By applying texture maps to the diffuse, specular, reflective, and bump channels, you could quickly take a shiny new car and make it look very weathered and old, and more importantly, more realistic for the various settings that your car will travel through. This, of course, holds true for all objects. Look closely at objects in the real world and you will immediately notice that very few of them have perfect, unblemished surfaces. Making your LightWave objects look imperfect is very important to adding realism in your work and using texture maps on the various surface channels is a step in the right direction toward achieving that realism.

On-Topic Quick Tip

Did you know that The Hub stores all of its data in a folder inside the Temporary system folder? If you have the Synchronize to Layout option activated in Modeler, then any changes you make to your object can be found in the Windows/Temp/lwhub folder (or the equivalent directory for Mac users). This can be very handy for retrieving modified objects if the system crashes before you have had a chance to save your work or if you make a change that is not undo-able. Take a look in that Temp/lwhub folder and you might be surprised by what you find there!

*David Warner is the creator of the TutorVision 3-D series of multimedia tutorial CD-ROMs for LightWave. He has also created a set of searchable archive CDs for the LightWave mailing list that give you quick and easy access to more than 100,000 questions & answers from other LightWave users around the world! For more info and a FREE Sampler CD, please visit **http://www.tv3d.com** or e-mail **dave@tv3d.com***

Final Cut (continued)

Continued from Page 64

you're solution-oriented, remember? You've already called the Systems department and they're restoring the object from a backup tape.

Rule Number Three: It's Only A Movie. Some people forget that fact, and get caught up in their own mythology. Don't let them get to you, and don't succumb to the same temptation. Try to have a life outside of computers, get some fresh air, and remember that we could all be back in the insurance business or selling cars next month. Sometimes I think that a year's work digging ditches or waiting tables should be a prerequisite for working in a business as cool as this one, just to make us remember how lucky we really are.

Rule Number Four: Try to Save the Bridge. OK, you've had your run-in with that nutter coworker, client, boss, or employee. Before you tell them to take a flying leap, ask yourself if there's a way to exit amicably. You never know, especially in this business, when that P.A. you're about to piss all over is going to wind up producing a major feature right when you're out of work. Granted, there will be times where the only way out of the situation is going to be a hearty "Bite Me!" but where possible, take the high road. It may not feel as good at the time, but you're in this for the long haul, right?

In the end, it all comes down to the Golden Rule: Do Unto Others and all that jazz. Play the game by rules that you can live with, and try to play honorably. With all the craziness out there, you'll stand out as someone who people enjoy working with, and that's how a job in movies turns into a career.

Steve Pugh is the General Manager at Foundation Imaging. You can contact him through its website, www.foundation-i.com

TUTOR VISION 3-D CD

The BEST WAY to learn LightWave!

Easier to understand than a book...more tutorials than any video-tape!

Check out our two newest releases!

TutorVision3-D Volume 5 is our first set of tutorials for the new LightWave v6.5!
Available NOW for **$49.95**

LightWave Mailing List Search Engine CD Year 2000 Edition
Available NOW for **ONLY $9.95!**

FREE Sampler CD!

This FREE SAMPLER CD introduces you to the TutorVision3-D series of CD-based video tutorials. This CD contains 1 complete video and text tutorial from each of the four currently available volumes of TutorVision! It also contains 2 brand new video tutorials not found on any other TutorVision3-D CD!

That's 10 video and text tutorials for **FREE!**

So how do you get your FREE Sampler CD?

Just get your address to us via e-mail, phone, or use the special form on our web-site, and we'll mail the Sampler CD to you right away, with no obligation to buy anything!

Quantities Limited - ORDER NOW!

NEW LOW PRICES on all of our "older" titles!!

We've made **BIG** price reductions on our first 5 CDs! Check out our web-site or give us a call for complete details and pricing!

Event Horizon Studios
Phone: 518-756-6850
FAX: 518-756-6836
E-Mail: info@tv3d.com
Web-Site: http://www.tv3d.com

Add a FREE issue...

when you subscribe online at www.keyframemag.com

Require an Invoice?

Print invoices online to qualify for the free issue.

Back issues...

also available online!

KEYFRAME MAGAZINE

DMG Publishing
2756 N Green Valley Pkwy # 261
Henderson, NV 89014
Tel 702-990-8656 Fax 702-616-9647
888-778-WAVE

☐ USA 1 year (6 issues)	$54.00
☐ Canada/Mexico 1 year	$72.00
☐ All other countries 1 year	$84.00
Or subscribe for two years!	
☐ USA 2 years (12 issues)	$102.00
☐ Canada/Mexico 2 years	$138.00
☐ All other countries 2 years	$162.00

☐ This is a new subscription ☐ This is a renewal

Name ______________________________

Company Name ______________________________

Address ______________________________

City, State and Zip ______________________________

Country ______________________________

Phone ______________________________

Fax ______________________________

Email ______________________________

☐ Charge my card (Visa, MasterCard, American Express, DiscoverCard)

Card # ______________________ Exp.________

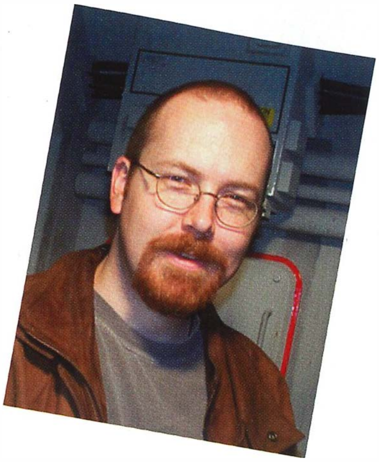
Name on credit card ______________________________

Signature (Required) ______________________________

☐ Check or Money Order payable to DMG Publishing enclosed. US Dollars only.

It Takes All Kinds

by Steve Pugh



Hooray for Hollywood! There you were a couple short years ago (maybe even scant months ago if you've got the chops), building your first cylinder in this program called LightWave. Visions of working in the movie business danced through your head. There you are on the set, conferring with the Director of Photography on how best to light Mr. Willis for your multi-layer composite. There you are, taking meetings with Steven and George, showing them the latest motion tests for the Space Weevil from Episode Three.

There you are, getting screamed at by some guy who's been in the business twelve seconds longer than you have, because he can smell your coffee from his desk. Next thing you know, there's a keyboard embedded in the wall next to your head.

What, you thought the movie business was sane?!? My friend, I have some news for you. A lot of us are completely screwed up.

I don't know what it is about this business, maybe it's the sacrifices that one has to make to succeed - twenty-hour days, seven day weeks, working on locations where it's either snowing, raining, hailing, or so hot that your sunburn is getting a sunburn. Maybe it's something about sitting in a dark room for hours on end staring at a 19" diagonal box which contains your entire world. Maybe the genius and passion required to take an idea from thought to screen are so intense that they actually change the psyche.

Either way, it's safe to say that if you spend enough time in the film world you're going to come across someone who's a couple polys short of a solid mesh. A few points short of a curve. The lights are on, but the scene renders black. You get the idea.

What do you do? Well, you do your best. Like any good soldier, you adapt, improvise, and overcome.

Rule Number One: Don't Stoop. When everyone around you is snapping like Pop Rocks in a can of soda, resist the temptation to join them. You may be working on the world's most hopelessly screwed film, but you're a professional. Be solution-oriented, and spend your energy finding options instead of looking for places to point your finger. Sure, if someone is trying to set you up as the reason why things are unglued, you certainly have the right to defend yourself. Just remember that a defense with a solution attached is going to get more done and make you look better than just digging your heels in and shouting "Nuh uh!!!"

Rule Number Two: Admit Your Mistakes. You just overwrote the Star Destroyer hero object with a checkered cube. The FX Supervisor is honing a straight razor while screaming "Who *@#&%'ed this up?!?" and you don't recall ever seeing him that shade of purple before. You could sink down into your chair and shrug, or you could 'fess up. Either way, someone's going to find out the truth. Would you rather it be after you deny it? Besides,

Continued on Page 62

Aura 2
The Desktop Video Paint System
Paint over video and bring life to your animations.

Terrence Walker, creator of **'Understanding Chaos'**, has already discarded all his other 2D and painting tools in favor of Aura's unique flexibility, **"I can't mess with three different packages to get my work done...This one only paints, this other one can composite but you need a third for true 2D animation... Who needs that? With Aura, it's all right in front of me and I can do whatever I need to get the shot done!"**

Whether you are working on print graphics, web animations or video productions, Aura 2 from NewTek is the tool of choice. Aura's next-generation paint engine boasts impressive new features including **four-point pixel tracking, interactive keyframer, stroke recorder, natural media brushes** and **unlimited layers** with **real-time alpha channel**. Enhanced **scripting power** cut hours off production time and adds **flexibility, simplicity** and **speed**. Walker says nothing he's used comes close to matching the speed he's found in Aura. **"Speed is a real issue when you're one person trying to make a real show... Aura is lightning fast!"**

Walker's animation, created in Aura, can be seen at
http://www.studioartfx.com
Read more about Aura at **http://www.newtek.com**

Creativity without limits is yours with Aura

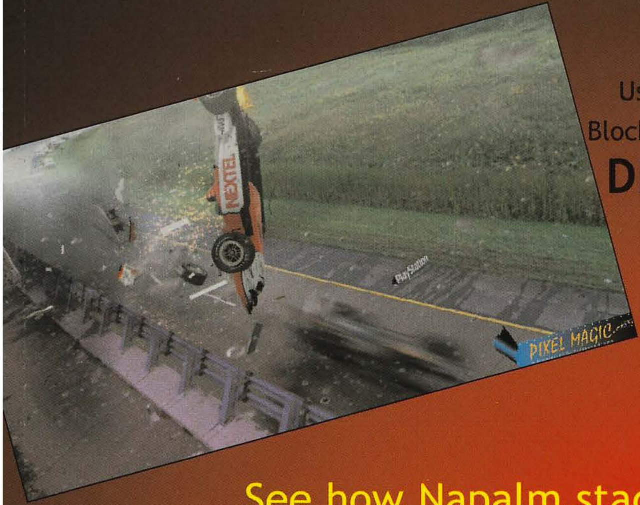
NewTek
8200 IH-10, Suite 900, San Antonio Texas 78230 Phone: 220-370-8000 Fax: 210-370-8001
www.newtek.com

Creative imagery by:
TERRENCE WALKER
www.studioartfx.com/uc_view.htm

NAPALM

PARTICLE STORM 3

LightWave's most powerful particle system just got better!



Used in the
Blockbuster Film
DRIVEN

PIXEL MAGIC

See how Napalm stacks up against the others!

Features	LightWave's PFX	Particle Storm 2 Pro	Napalm/Particle Storm 3
Integrated into LightWave	Yes	No	Yes
Reacts to deformations, including Messiah & MD2K	No	No	Yes
Use polygons as particles	No	No	Yes
Remove a particle for a portion of scene	No	No	Yes
React and effect with Impact 2 objects	No	No	Yes, particles can affect impact objects
Force Fields, Drag	Yes, but only thru the Wind controller	Yes, Yes	Yes, Yes
Support for Weight Maps, Pick Maps, and Parts	No	No	Yes, throughout the program
Envelopable thru the Graph Editor	Yes, but only a few things	No	Yes, everything can be enveloped and controlled by channel modifiers
Can be controlled by expressions	Yes, but only a few things	No	Yes, everything can be tied to expressions
Gives the user godlike control over the particles	No	No	YES!!!

Dynamic Realities
1999 Whitney Mesa Drive, Ste G
Henderson, NV 89014
(702) 990-2277 or (800) 944-5520

For more information on Napalm or our other products, visit our web site at:
www.dynamic-realities.com